НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повназваінституту/факультету)

Автоматизованих систем обробки інформації і управління
(повназвакафедри)

«На правах рукопису»
УДК 004.93

До захисту допущено:
В.о. завідувача кафедри
_____ Олександр ПАВЛОВ
«___»_____20__ р.

Магістерська дисертація

на здобуття ступеня магістра

за освітньо-професійною програмою «Інженерія програмного забезпечення комп'ютеризованих систем»
зі спеціальності 121«Інженерія программного забезпечення»

на тему: «Розпізнаванняголосових команд управліннядля пристроїв на базіArduino»

Виконав (-ла):
студент (-ка) VI курсу, групи ІП-91мп
Закупін Єгор Олександрович _____

Науковийкерівник:
доц., к.т.н.
Мажара Ольга Олександрівна _____

Рецензент:
доц. каф. АПЕПС ТЕФ, к.т.н., доц.
Шаповалова Світлана Ігорівна _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент (-ка) _____

Київ – 2020 року

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальноїтехніки
Автоматизованих систем обробкиінформації і управління

Рівеньвищоїосвіти – другий (магістерський)
Спеціальність – 121 «Інженерія програмного забезпечення»
Освітньо-професійнапрограма - «Інженерія програмного забезпечення комп'ютеризованих систем»

ЗАТВЕРДЖУЮ
В.о. завідувача кафедри
_____ Олександр ПАВЛОВ
«___»_____20__ р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Закупіну Єгору Олександровичу

1. Тема дисертації «Розпізнавання голосових команд управління для пристроїв на базі Arduino», науковий керівник дисертації Мажара Ольга Олександрівна, доцент, к.т.н., затверджені наказом по університету від «___»_____ 20__ р. №_____

2. Термін подання студентом дисертації: _____

3. Об'єкт дослідження: розпізнавання звукових команд словника з використанням обмежених апаратних характеристик

4. Вхідні дані: набір аудіо-записів звукових команд

5. Перелік завдань, які потрібно розробити: Дослідити існуючі розпізнавачі команд на базі Arduino, провести порівняльну характеристику існуючих рішень, та обґрунтувати необхідність створення їх аналогу, виділити основні функції системи розпізнавання, описати алгоритм, описати розробку программного забезпечення, привести результати тестування та приклади роботи

6. Орієнтовний перелік графічного (ілюстративного) матеріалу: схема алгоритму, загальна схема роботи проекту

7. Орієнтовний перелік публікацій: Закупін Є.О. РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД УПРАВЛІННЯ ДЛЯ ПРИСТРОЇВ НА БАЗІ ARDUINO/ Закупін Є.О., Мажара О.О. // Матеріали V всеукраїнської науково-практичної конференції молодих вчених та студентів «Інформаційні системи та технології

управління» (ІСТУ-2020) – м. Київ: НТУУ «КПІ ім. Ігоря Сікорського», 26-27 листопада 2020р.

8. Консультанти розділів дисертації

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
Графічний	доц. Ліщук К.І.		

9. Дата видачі завдання_____

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1	Затвердження теми роботи	15.09.2020-20.09.2020	
2	Вивчення та аналіззавдання	20.09.2020-30.09.2020	
3	Розробкаархітектури та загальної Структури cucmem	31.10.2020-23.11.2020	
4	Оформленняпояснювальної записки	24.11.2020-26.11.2020	
5	Передзахист	27.11.2020	
6	Захист	17.12.2020	

Студент Єгор Закупін

Науковий керівник Ольга Мажара

РЕФЕРАТ

Магістерська дисертація: 85 с., 18 рис, 28таб., 12 джерел.

Актуальність теми: Забезпечення якості життя людей з обмеженими можливостями є актуальною проблемою як в усьому світі, так і в Україні зокрема. Провідними країнами створюються новітні розробки для спрощення побуту таких людей. Нажаль, в Україні таким розробкам не надається значної уваги. В той же час, використання надбань зарубіжних колег часто обмежується фінансовим та/або мовним бар'єром. Необхідними є розробки орієнтовані на локальний ринок та потреби.

Мета дослідження: розробити систему розпізнавання голосових команд управління для пристроїв на базі Arduino.

Для реалізації поставленої мети були сформульовані **наступні завдання:**

- Дослідити існуючі розпізнавачі команд на базі Arduino.
- Провести порівняльну характеристику існуючих рішень, та обґрунтувати необхідність створення їх аналогу.
- Виділити основні функції системи розпізнавання.
- Описати алгоритм.
- Описати розробку программного забезпечення
- Привести результатитестування та приклади роботи.

Об'єкт дослідження: розпізнавання звукових команд словника з використанням обмежених апаратних характеристик.

Предмет лослідження: опис програмних алгоритмів та ознак, які використовуються при розпізнаванні та класифікації звукових команд.

Методи дослідження: структурно-функціональні методи в дослідженні нейронних мереж.

Наукова новизна: найбільш суттєвими науковими результатами магістерської дисертації є:

- розробка нової бібліотеки машинного навчання для використання у проектах для розпізнавання мови в умовах обмежених апаратних ресурсів.

Практичне значення отриманих результатів визначається тим, що була створена система, що дозволяє ефективно то точно виконувати управління інвалідним візком за допомогою голосових команд управління.

Зв'язок роботи з науковими програмами, планами, темами: робота виконувалась на кафедрі автоматизованих систем обробкиінформації та управління Національного технічного університету України «Київський політехнічний інститут ім. Ігоря Сікорського» в рамках теми «Моделі та технології інтелектуальних обчислень».

Апробація: Основні положення роботи доповідались і обговорювались на V всеукраїнській науково-практичній конференції молодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020).

Публікації: Наукові положення дисертації опубліковані в Закупін Є.О. РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД УПРАВЛІННЯ ДЛЯ ПРИСТРОЇВ НА БАЗІ ARDUINO/ Закупін Є.О., Мажара О.О. // Матеріали V всеукраїнської науково-практичної конференціїмолодих вчених та студентів «Інформаційні системи та технології управління» (ІСТУ-2020) – м. Київ: НТУУ «КПІ ім. Ігоря Сікорського», 26-27 листопада 2020р.

Ключові слова: НЕЙРОННІ МЕРЕЖІ, РОЗПІЗНАВАННЯ МОВИ, ОПТИМІЗАЦІЯ

ABSTRACT

Master's dissertation: 85pp., 18 figures, 28 tab., 12 sources.

Relevance of the topic: Ensuring the quality of life of people with disabilities is an urgent problem both around the world and in Ukraine in particular. Leading countries are creating the latest developments to simplify the lives of such people. Unfortunately, in Ukraine such developments are not given much attention. At the same time, the use of foreign colleagues' assets is often limited by financial and/or language barriers. Developments focused on the local market and needs are needed.

The purpose of the study: to develop a system for recognizing voice control commands for devices based on Arduino.

To achieve this goal, the **following tasks** were formulated:

• Explore existing Arduino-based command recognizers.

• Carry out a comparative description of existing solutions, and justify the need to create an analogue.

• Highlight the mainfunctions of the recognition system.

• Describe the algorithm.

• Describes of tware development

• Give test results and examples of work.

Object of study: recognition of vocabulary sound commands using limited hardware characteristics.

Subject of research: description of software algorithms and features used in the recognition and classification of sound commands.

Research methods: structural and functional method sinthestudy of neural networks.

Scientific novelty: the most significant scientific results of the master's dissertation are:

- development of a new machine learning library for use in language recognition projects in conditions of limited hardwarere sources.

The practical significance of the obtained results is determined by the fact that a system has been created tha tallowsy outo effectively and accurately control the wheel chair with the help of voice control commands.

Relationship with working with scientific programs, plans, topics: the work wasper formed at the Department of Automated Information Processing and Control Systems of the National Technical University of Ukraine "Igor Sikorsky Kyiv Polytechnic Institute "in the framework of the topic" Models and technologies of intelligent computing ".

Approbation: The main provisions of the work were reported and discussed at the V All-Ukrainian scientific-practical conference of young scientists and students "Information systems and management technologies" (ISTU-2020).

Publications: Scientific provisions of the dissertation are published in Zakupin Ye. O. RECOGNITION OF VOICE CONTROL COMMANDS FOR ARDUINO-BASED DEVICES / ZakupinYe.O., Mazhara OO // Proceedings of the V All-Ukrainian Scientificand Practical Conference of Young Scientists and Students "Information Systems and Management Technologies" (ISTU-2020) - Kyiv: NTUU "Igor Sikorsky Kyiv Polytechnic Institute ", November 26-27, 2020.

Keywords: NEURAL NETWORKS, LANGUAGE RECOGNITION, OPTIMIZATION

ВСТУП

Даний проект показує принцип проектуваннясистеми для розпізнавання голосових команд управління для пристроїв на базі Arduino. Необхідністьвирішенняпоставленоїзадачігрунтується на розвиткуінформаційнихтехнологій, які у свою чергудають можливість покращувати та полегшувати побут, особливо для людей, що в цьому обмежені. Таким чином утворюється можливість інтеграції машинного навчання у побут людей з обмеженими можливостями, а саме – утворення системи голосового управління інвалідним візком.

Актуальністьроботиполягає у тому, щоіснуючірішеннямають низку недоліків, середяких: малий розмір словника для систем управління, спирання на застаріліверсіїапаратних засобів, рішення є сутокомерційним проектом а також широка спеціалізація, що ускладнює їх використання в локальному випадку. Тому ціллює створеннясистеми управління, яка б дозволяла використовуючи программу для навчанняотримуватифайлинавчаноїсистеми, що в свою чергубудутьвикористовуватись для апаратноїскладової на базі Arduino. Система буде навчатись не на існуючій вибірці, а буде використовувати записи, що дозволяє використовувати зручну мову для управління, а також полегшувати розпізнавання, використовуючи голос користувача.

Згідно з зазначеноюціллюрозробкиотриманінаступнізадачі:

- Дослідити існуючі розпізнавачі команд на базі Arduino.
- Провести порівняльну характеристику існуючихрішень, та обґрунтуватинеобхідністьстворенняїх аналогу.
- Виділитиосновніфункціїсистеми розпізнавання.
- Описати алгоритм.
- Описатирозробкупрограмногозабезпечення
- Привести результатитестування та прикладироботи.

Новизноюданного проекту є утворення в ході роботи нової бібліотеки для розпізнавання голосових команд, що може бути використана у будь-якому іншому проекті у середовищі для розробки проектів Arduino. Ця бібліотека може бути включена до проекту як будь-яка інша бібліотека мови Arduino. Вона надає змогу використовувати отриману навчану систему для розпізнавання голосу, що є більшефективною та економною в плані ресурсів порівняно з іншими видами реалізації.

Практичне значення проекту є покращення побуту для людей о обмеженими можливостями. Данне рішення надає змогу користуватися лише голосовими командами при управлінні інвалідним візком, що має велике значення особливо для людей, для яких існують складнощі при використовуванні ручного управління.

Наукові положення дисертації опубліковані в Закупін Є.О. РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД УПРАВЛІННЯ ДЛЯ ПРИСТРОЇВ НА БАЗІ ARDUINO/ Закупін Є.О.,Мажара О.О. // Матеріали V всеукраїнської науково-практичної конференціїмолодих вчених та студентів «Інформаційні системи та технології управління»(ІСТУ-2020) – м. Київ: НТУУ «КПІ ім. Ігоря Сікорського», 26-27 листопада 2020р.

У першому розділі розглянуті та досліджені існуючі розпізнавачі команд на базі Arduino, такожпроведена порівняльна характеристика існуючих систем, та обґрунтована необхідність створення їх аналогу. Це порівняння надає можливість виділити загальні риси подібних систем розпізнавання, а також допомагає визначити переваги та недоліки певних функцій та засобів. Такожвиділеніосновніфункціїсистеми розпізнавання, щонадаєбільшточнепредставлення про структуру системи для вирішенняпоставленоїзадачі.

У другому розділіописані методи та алгоритми для вирішення поставленої задачі. Представлені існуючі засоби та обгрунтований вибір обраного методу. Алгоритм надаєповну картину роботисистеми розпізнавання, а такожпоказуєособливостіструктурисистеми. У описі детально розглянутіскладові алгоритму, їхпризначення, та взаємодія.

У третьомурозділі описана розробкапрограмногозабезпечення. Показаніметодиреалізації алгоритму, а такожвиділеніотриманіфункціональнівластивості. Далінаведенірезультатитестуванняпрограми, у якихпроаналізованішвидкість та коректністьроботипрограми.

У четвертому розділі приводиться опис стартап проекту. Опис дозволяє побачити переваги обраного підходу, його можлівості при реалізації та потенційні покращення та збільшення єфективності розвитку стартапу.

Вся робота передбачає опис системи для розпізнавання голосових команд управління для пристроїв на базі Arduino, а також його програмна та апаратна реалізація, що надасть змогу у повній мірі оцінити розроблену систему, а також продемонструвати реалізовані функції, їх коректність, зручність та ефективність.

РОЗДІЛ 1 ДОСЛІДЖЕННЯ ІСНУЮЧИХ РОЗПІЗНАВАЧІВ ГОЛОСУ

1.1. Загальнийоглядіснуючихрішень

Згідно з поставлеою задачею були розглянуті існуючі методи розпізнавання командна базі Arduino. У загальному випадку існуючі рішення пожна розділити на 3 види.

Використання готових модулів для розпізнавання команд

Використання апаратної частини на базі Arduino у зв'язці з іншою технікою, на якій і буде проводитись розпізнавання (Комп'ютер чи Телефон)

Використання самого контроллеру для розпізнавання

Через обмежені ресурси контроллеру стає неможливо навчати систему на самому контроллері, тому попередньо до використання контроллеру у роботі з розпізнаванням команд система має буди окремо навчена, а вже потім отримані навчані файли будуть додані до контроллеру.

1.2. МашинненавчанняWekinator та Arduino UNO

ВикористовуєпрограмуmашинногонавчанняWekinator та Arduino UNO, щобкеруватикольоромсвітлодіода RGB своїм голосом.

Круговадіаграма

ПідключаєтьсянайдовшаніжкасвітлодіодаRGB до заземленняArduino. Іншініжкипідключаются до виводів 9, 10 і 11 Arduino через 220- омний резистор, як показано на схемінижче.

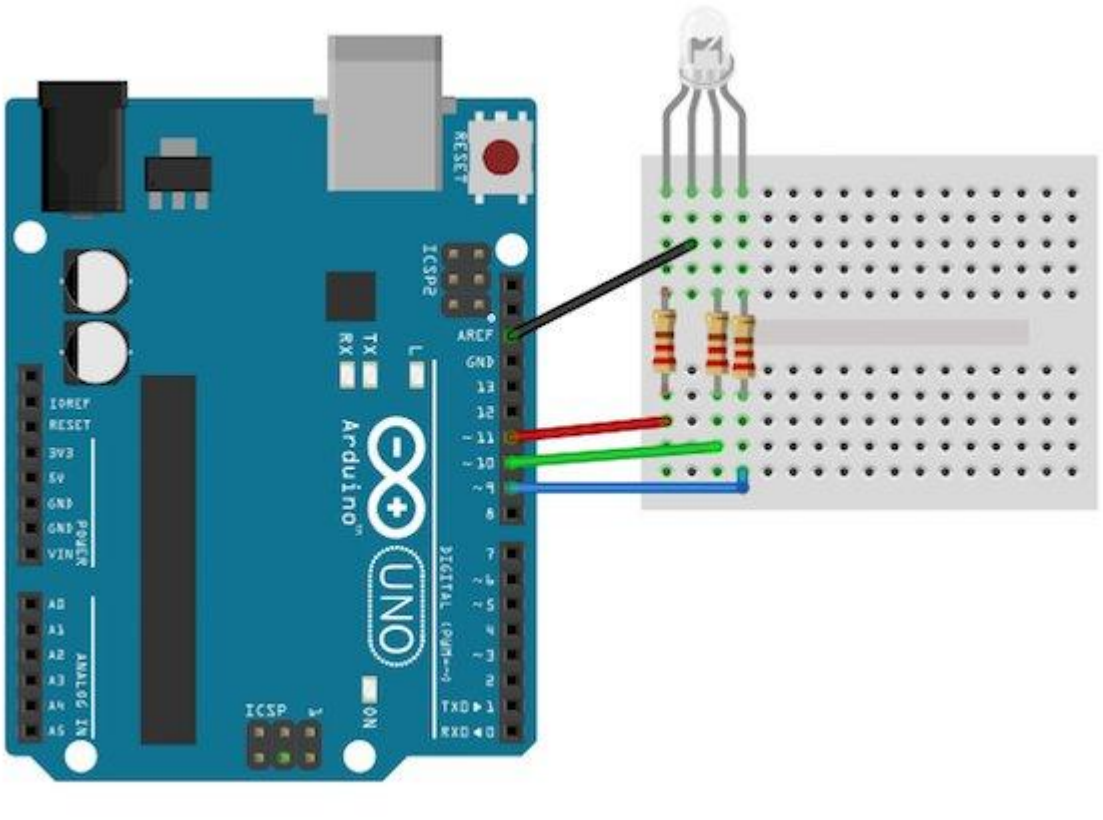

Рис 1.1. електрична схема

Як запуститипрограму

Перш за все, вставляется код, вказаний для Arduino, в IDE Arduino та завантажуєтсякод.

Потімпотрібнозавантажитиескіззі сторінкиприкладівWekinator .

Завантажуєтсявиконуваний файл для MFCC (цепстральнікоефіцієнти мел-частоти).

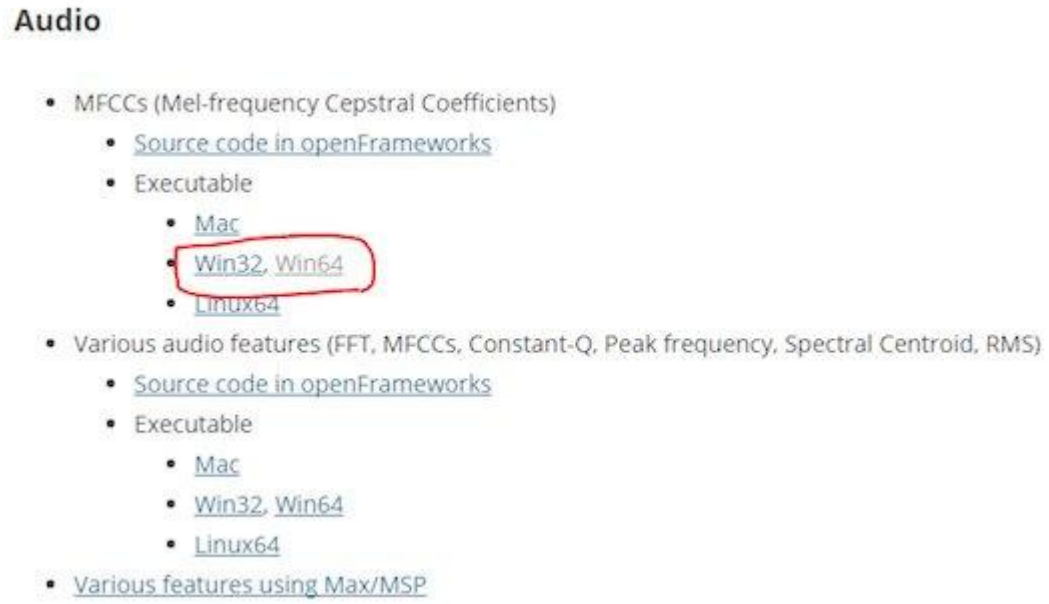

Рис 1.2. положення бібліотеки

Післязавантаженнярозпаковується та запускається файл .exe. Це буде виглядати так, як показано нижче. Теперпотрібенмікрофон, щоб подати вхід до Wekinator. Якщопідключаєтьсязовнішніймікрофон, необхіднощобвін буввибраний у налаштуваннях звуку на комп'ютері.

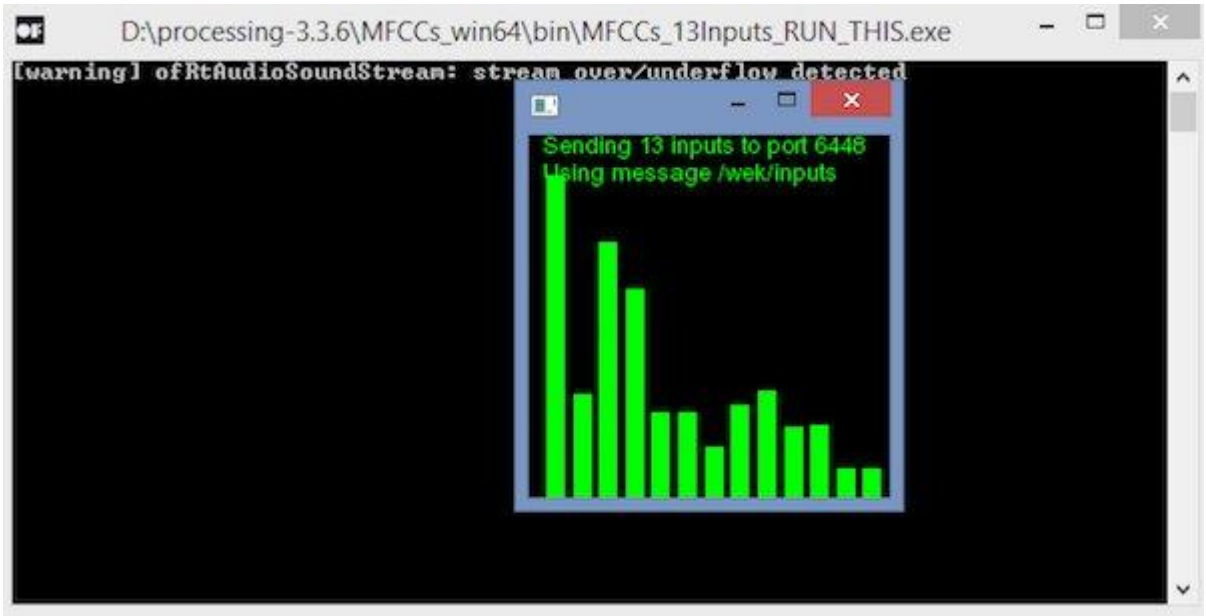

Рис. 1.3. Запуск програми

Такожпотрібенще один ескіз ("вихіднийескіз"), щоботриматывихіднідданівідWekinator. Необхідно вставитийого у новевікнообробки та запуститиескіз.

Тепернеобхідно відкритиWekinator і виконатиналаштування, як показано на малюнкунижче. Встановити входи на 13, а виходи на 1. Встановити тип на «Усідинамічні перекоси часу» за допомогою 3 типівжестів і натиснути на «далі».

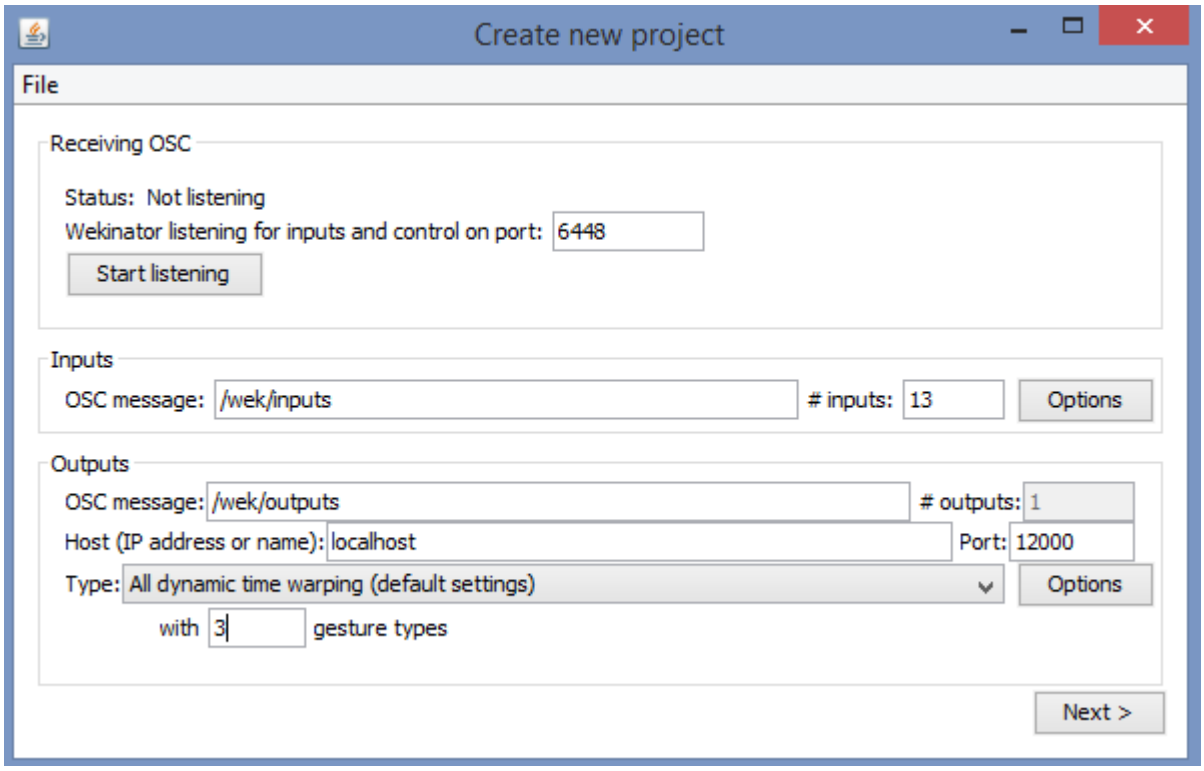

Рис. 1.4. Вікно налаштувань

Тепертреба утримувати кнопку “+” перед output_1 і сказати “червоний”, для _2 - “зелений”, _3 - “синій”.

Післяцьогонатиснути «train», потімнатиснути «Виконати». Теперколірсвітлодіода RGB змінюватиметьсявідповідно до названогокольору.

1.3. Голосовеуправління за допомогоюArduino та Android

Як працюєуправління голосом?

Сама схема досить проста, її основою служить мікроконтролерArduino, який і буде обробляти сигнал від сенсора, виробляючи в залежностівідкоманди, за допомогоювиконуючоїчастини, необхіднудію. Вибірсамейого - орієнтованість і модульність. Віннайбільшповновписується в поняття «розумнийбудинок», дозволяючирозширюватисвоїможливості практично

безмежноюкількістюзовнішніхмодулів, зручною системою програмування і інтерфейснимвзаємодією з комп'ютером.

Принципифункціонування самого розпізнавання голосу, а відповідно і вид сенсора. Їхдекілька:

икористовуватизв'язок з online - службами обробки голосу, на зразокGoogleабоYandex, з подальшоюреакцією контролера на переданірезультати, причому в такому випадку буде використовуватисябільш «розумне» пристрій - посередник, комп'ютерабо смартфон;

розпізнаванняголосових фонем силами самого мікроконтролера;

обробказвуковихсигналівдодатковим модулем, щопідключається до Arduino.

Зрозуміло, що в першомувипадкунаявністьпідключеногомікрофона до самої плати контролера не потрібно. Досить буде або прямого приєднанняінтерфейснихпроводіввідкомп'ютера, абовикористаннядодатковогоBluetooth - модуля для забезпеченнязв'язку в разі смартфона. Останнійваріант і буде розглянуто.

Представлена схема голосового управління на базіАрдуіно буде демонстративною, максимально швидкоговиготовлення. Але на їїосновіжеможнастворюватиреальнісистемиобробкиголосових команд.

Обрана для демонстраціїможливостейуправління голосом за допомогоюАрдуіно схема для виготовленнясвоїми руками буде запалювати, в залежностівідкоманди, світлодіодпевногокольору.

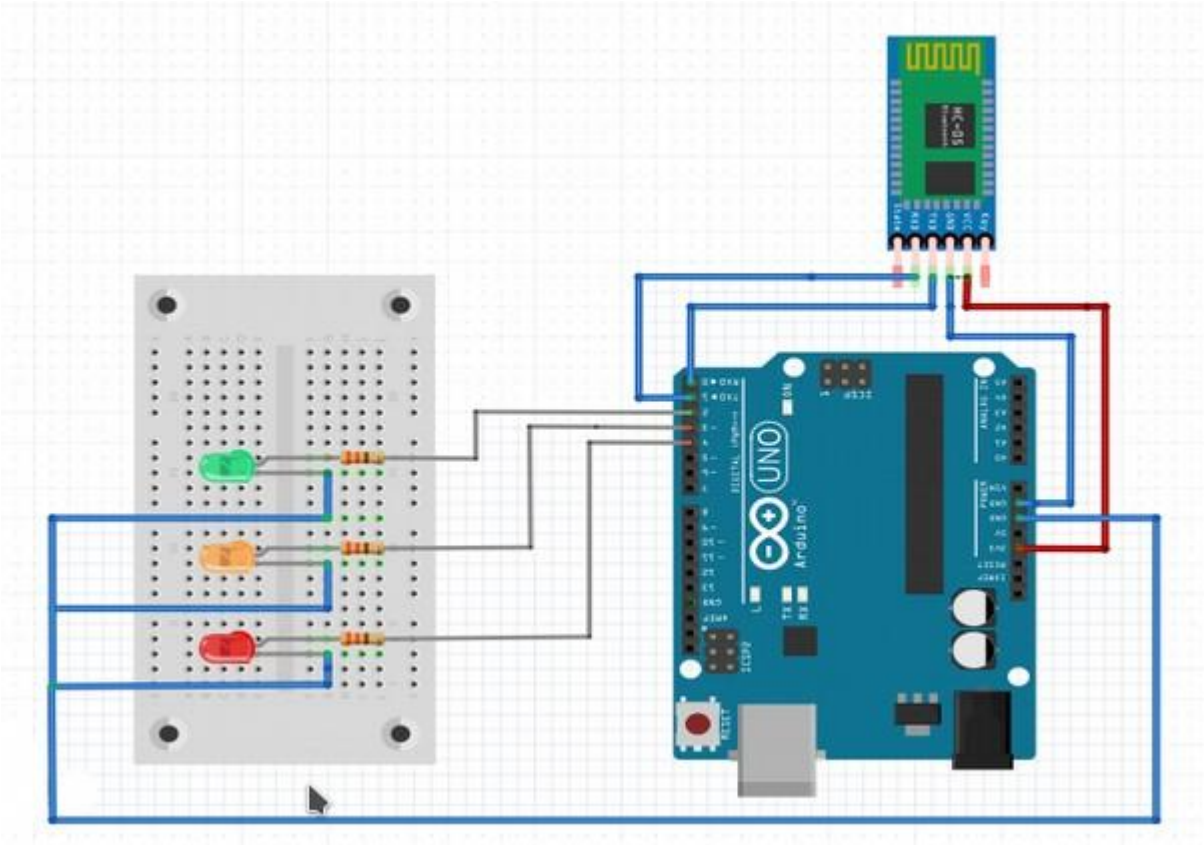

Рис. 1.5. Електрична схема

Інструменти і матеріали

Для складаннясистемиуправління голосом необхіднонаступне:

сама плата мікроконтролераArduino UNO;

міні - схема HC-05, на якійрозлучений модуль зв'язкуbluetooth для Ардуіно;

три резистора на 300-350 Ом;

світлодіодирізнихкольорів - 3 шт .;

смартфон.

Порядок підключеннякомплектуючих

Спочаткунеобхіднопідключитимодульну плату, щозабезпечуєзв'язок по bluetooth, до Ардуіно. Контакт RXD від HC-05 приєднуємо до клеми TXD контролера, аналогічну модуля, з RXD Arduino. Тобто, виробляємоперехреснепідключенняроз'ємів. Вхід +3.3 і GND на збірціbluetoothпідводиться до аналогічних на платі контролера.

Далі, підключаємовиконавчий блок, управління голосом яким і буде здійснюватися через Ардуіно. Спільнийвміст «землі» (GND) діодівз'єднуємо з

таким же на центральнійплаті. Потімкожен з світлодіодів, другийніжкою, через резистор на 330 Ом з вихіднимицифровими каналами контролера (2-4).

Сама діалоговачастина з користувачем - додаток на смартфоні. Йогоможназнайти в гугл-маркет по «BT VOICE CONTROL для ARDUINO» (BT VOICE CONTROL FOR ARDUINO) від SIMPLELABSIN.

Комунікаціязробленоїсхеми і смартфона проводиться при поданому на неїнавантаженні і виборі в меню програми BT VoiceControl (верхнійправий кут) «підключити робота». Відкриєтьсявікнозі списком комунікаційногообладнання - в ньомупотрібнознайти і вибрати HC-05.

От і все. Можнапрацювати. Командивикористовуютьсяті, якіприписані в коді скетчу. Наприклад, сказане голосом «POWER ON GREEN» - включить зеленийсвітлодіод.

Особливості настройки Arduino для голосового управління

Так як розробникросійськомовний, команди, щовіддаються на іноземніймові - не дуже хороша ідея, навіть для демонстрації. Можназамінитиїх на латиницю, де російський звук пишетьсяанглійськимилітерами. Наприклад, якщо рядок скетчу

Elseif (voice == «* poweronyellow»)

замінити на

Elseif (voice == «* vkluchitgeltuy»)

то можна буде використовуватиголосову команду «включитижовтий». На жаль, звучаннялатинськогоалфавітутрохивідрізняєтьсявідросійського, тому тут потрібнопоекспериментувати.[1]

1.4. ElechouseVoiceRecognitionModule v3.1: модуль розпізнавання голосу

Розглянемо один з найпростішихспособівнавчитиArduinoрозумітиголосовікоманди - ElechouseVoiceRecognitionModule. Як приклад -

управліннясвітлодіодом. Голосова команда зможейоговключити, вимкнутиабозмуситиблимати.

Для роботипотрібно:

ElechouseVoiceRecognitionModule v3.1: модуль розпізнавання голосу

Плата Arduino. Наприклад, Arduino UNO R3 .

Будь-світлодіод і відповідний резистор для нього.

Arduino IDE зівстановленоюбібліотекою VoiceRecognitionV3

підключення

Підключення модуля дужепросте. Всього два Піна: TX і RX. Їх треба підключити до виходівArduino 2 і 3 відповідно. Живити модуль слідвід 5V.

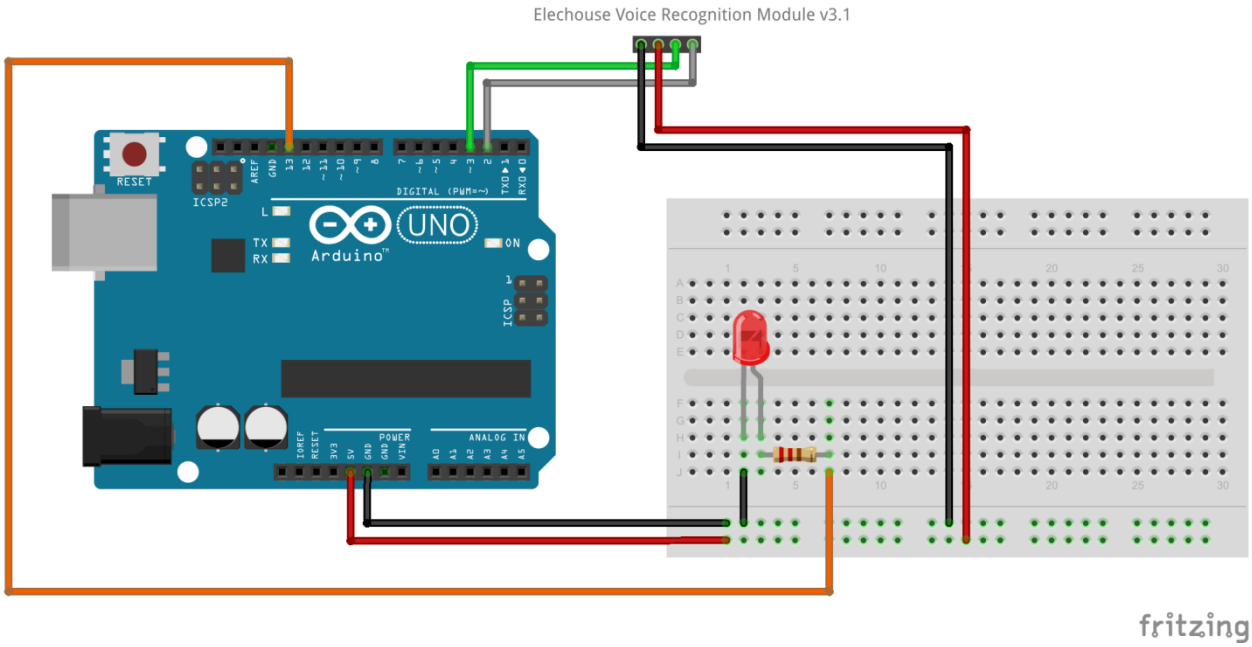

Рис. 1.6. Електрична схема

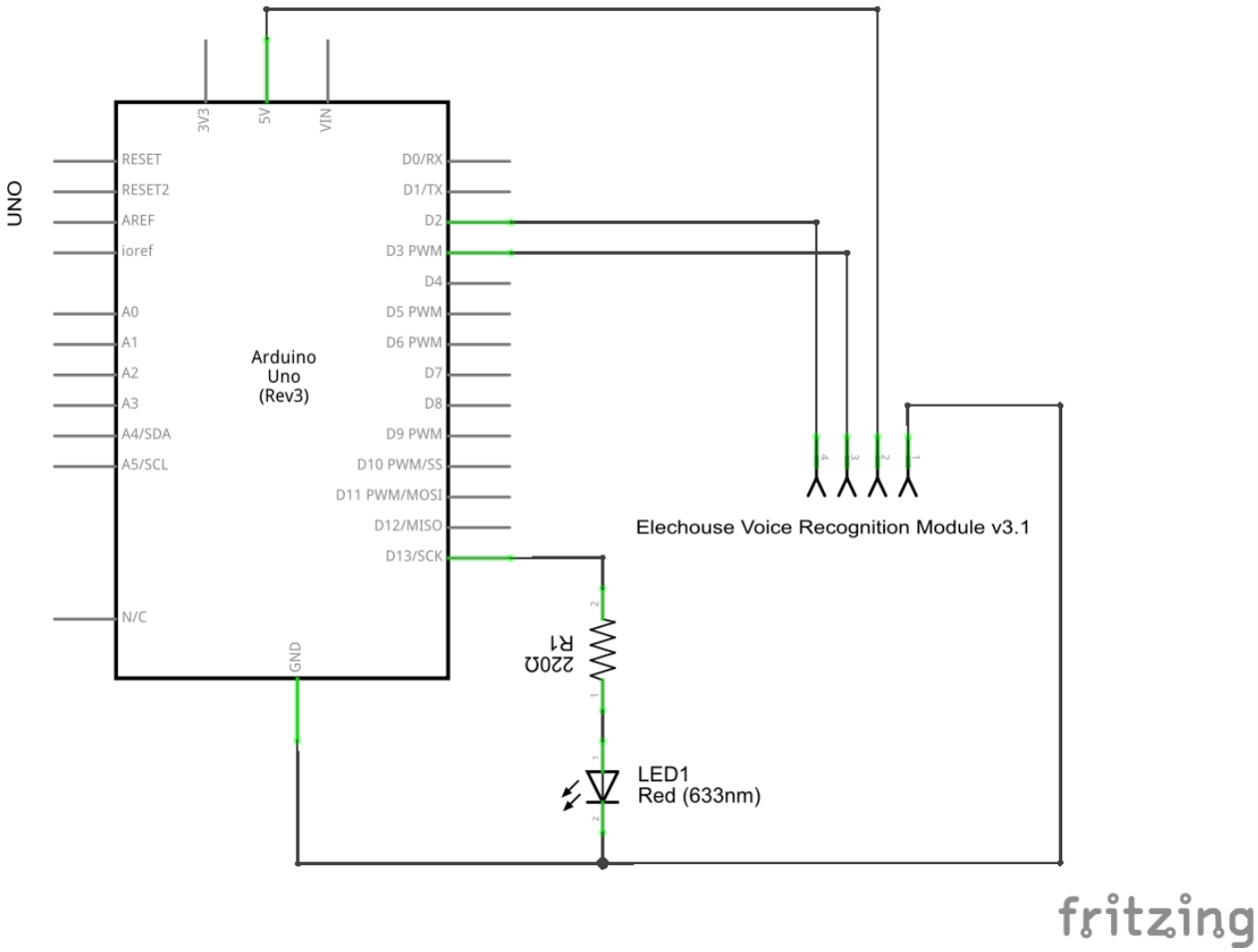

Рис. 1.7. Аналогова електрична схема

Навчання командам

Отже, насамперед треба наш модуль навчити командам. Як було сказано вище, всього три команди:

засвітись

вимкнути

Блимай

Відкрити проект **vr_simple_train** , поставляється в комплекті з бібліотекою VoiceRecognitionV3.

Файл - Приклади - VoiceReocgnitionV3 - vr_simple_train

Залийтицей скетч в Arduino і відкрийтиМонітор порту (**Сервіс - Монітор порту** абонатисніть на клавіатурі) **Ctrl** + **Shift** + **M**

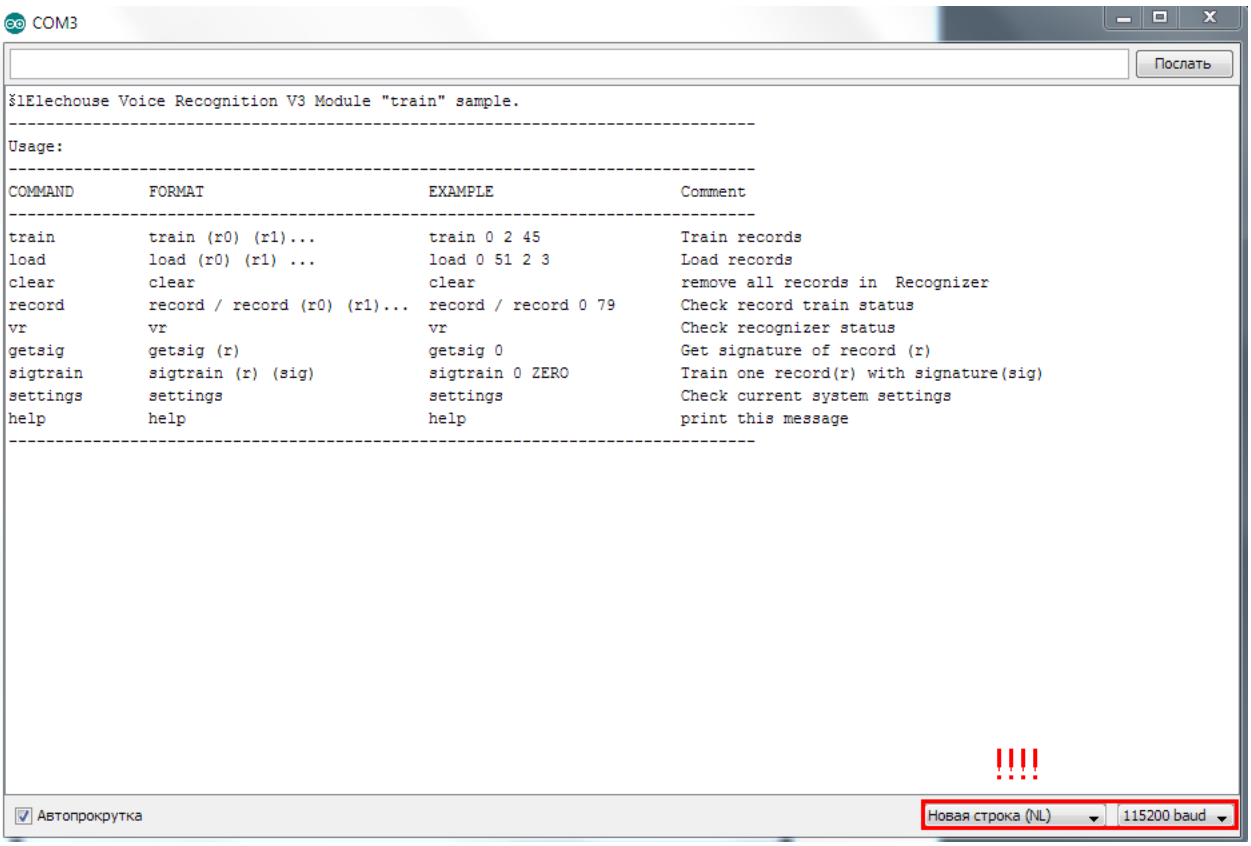

Рис. 1.8. діалогове вікно

Обов'язково треба виставитишвидкістьобміну (baudrate) 115200 і відправку по новому рядку.

Ввести в верхнє поле **settings** і натиснути кнопку " **Послати** ".

Модуль відповістьсвоїмипоточними настройками. Цеозначає, щоможнаприступатибезпосередньо до навчання команд.

За навчання командам відповідаєфункція **sigtrain** .

Необхідно ввести в поле команду **sigtrain 0 On** і натисніть на кнопку "Відправити". Команда означає, що в коміркупам'яті **0** требазаписати команду з сигнатурою **On** . Сигнатура цеякийсьунікальнийярлик, якийописуєкоманду.

Коли у вікніз'явиться фраза " **Speaknow** ", то слідпроговорити в мікрофон команду "Запалися".

Післяпоявифрази " **Speakagain** ", проговорити фразу ще раз.

Якщообидва слова збіглися, то модуль видасть **Success: 1** ,щоозначає, щокоманда **On** успішно саписана.

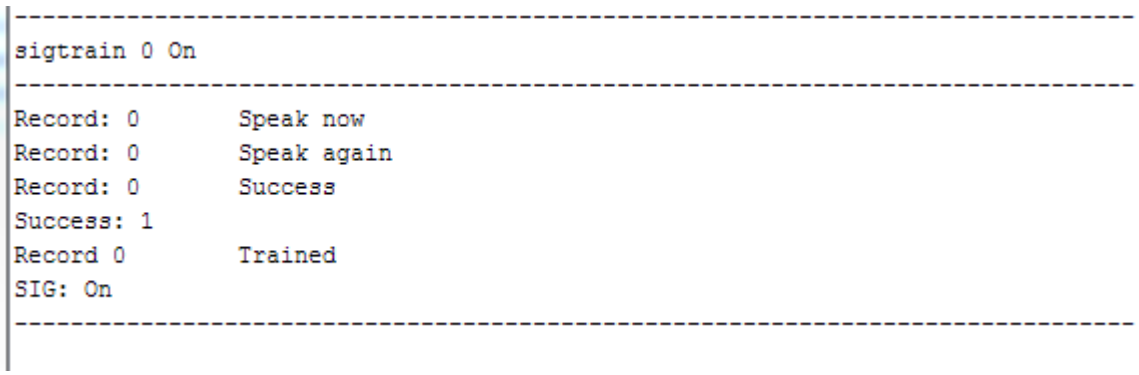

Рис. 1.9. Вивід програми

Якщо ж модуль не змігзіставитидвіголосовихкоманди (наприклад, було шумно в приміщенніабовимовлялись просто різні слова), то модуль відповість фразою "Cann'tmatch" і запропонуєпочатипроцесзаписукомандище раз до тих пір, поки не будутьнаданівірнідані

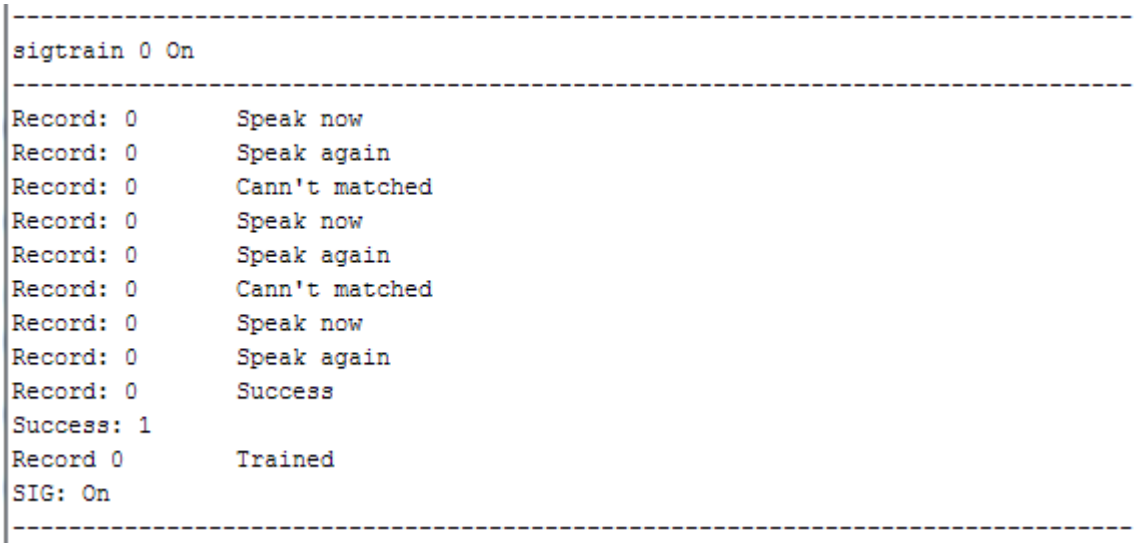

Рис. 1.10. процес розпізнавання.

Тежсаме треба зробити і з іншими командами "вимкнути" і "Мигай", але використовувативсііншіосередкипам'яті (1 і 2) і іншісигнатури (Off і Blink)[2]

Модуль розпізнавання голосу V3

Огляд

Модуль розпізнавання голосу ELECHOUSE - це компактна та проста в управлінні плата розпізнаваннямовлення.

Цейвиріб є модулем розпізнавання голосу, щозалежитьвіддинаміка. Вінпідтримує до 80 голосових команд. Макс. 7 голосових команд

можутьпрацюватиодночасно. Будь-який звук можнанавчитикоманді. Користувачампотрібноспочаткунавчити модуль, перш ніждозволитийомурозпізнавати будь-яку голосову команду.

Ця плата має 2 способиуправління: послідовний порт (повнофункціональний), загальнівхідніштифти (частинафункції). Загальнівихідніштифти на платіможутьгенеруватикількатипівхвиль при розпізнаваннівідповідноїголосовоїкоманди.

Особливість

Підтримка максимум 80 голосових команд, кожен голос 1500 мс (однеабо два слова)

Максимум 7 голосових команд, щодіютьодночасно

ПоставляєтьсябібліотекаArduino

Простеуправління: UART / GPIO

КеруваннякористувачемЗагальнийвихідний контакт

Термінологія

VR3 - модуль розпізнавання голосу V3

Розпізнавач - контейнер, кудизавантажувалисьголосовікоманди (максимум 7). Цеосновначастина модуля розпізнавання голосу. Наприклад, цепрацює як “гра в м’ячі”. У вашійкоманді 80 гравців. Але ви не могли дозволитиїмусімграти на корті разом. Правило дозволяєлише 7 гравців, якіграють на майданчику. Тут впізнавач - це список, якийміститьіменагравців, щопрацюють на майданчику.

Індексрозпізнавання - у розпізнавачіможепідтримуватися не більше 7 голосових команд. Розпізнавачмає 7 областей для кожноїголосовоїкоманди. Один індексвідповідаєоднійобласті: 0 ~ 6

Поїзд - процесзапису ваших голосових команд

Завантажити - скопіюватинавчений голос у розпізнавач

Записголосових команд - навчений магазин голосових команд у спалаху, номер від 0 до 79

Підпис - текстовийкоментар для запису

Група - допомога в управліннізаписами, кожнагрупа по 7 записів. Підтримуються системна група та групакористувачів.

Протокол VR3

Протокол VR3 міститьосновнікоманди для управління плат VR3. Для тих, хтовикористовує VR3 з іншими MUC, а не з Arduino, протокол VR3 дужекорисний.

Всікоманди VR3 надсилаються через послідовний порт у ГЕКСАДЕКЦІМАЛЬНОМУ ФОРМАТІ.

Прикладипостачаються з цимінструментомпослідовного порту: Порт доступу

Щобпідключити VR3 до ПК, цейінструмент модуля USB-TTL: Модуль USB-TTL на 5 В або 3,3 В

Базовий формат

Контроль

| *Голова (AA)* | *Довжина* | *Командування* | *Дані* | *Кінець (0A)* |

Довжина = L (довжина + команда + дані)

Повернення

| *Голова (AA)* | *Довжина* | *Командування* | *Дані* | *Кінець (0A)* |

Довжина = L (довжина + команда + дані)

ПРИМІТКА. Область данихвідрізняєтьсявідрізних команд.

Код

ВСІ КОДИ В БЕЗ ШЕСТИГРАДНИЧНОМ ФОРМАТІ КАДРОВИЙ КОД

AA -> Головка кадру

0A ->Кінець кадру

ПЕРЕВІРКА

->Перевіртеналаштуваннясистеми

->Перевіртерозпізнавач

->Перевірте стан записупоїзда

->Перевіртепідпис одного запису

НАЛАШТУВАННЯ СИСТЕМИ

->Відновитиналаштуваннясистеми

->Встановитишвидкістьпередачі

->Встановитивихідний режим вводу-виводу

->Встановитивихідну ширину імпульсу вводу-виводу

->Скинутивихідний IO

->Встановитиавтоматичнезавантаження

ОПЕРАЦІЯ ЗАПИСУ

->Тренуйте один записабо записи

->Навчіть один запис і встановітьпідпис

->Встановитипідпис для запису

КОНТРОЛЬ РОЗПІЗНАВАЧА

->Завантажтезаписабо записи до розпізнавача

->ClearRecognizer

->Груповий контроль

ЦІ 3 КОДИ ВИКОРИСТОВУЮТЬСЯ ТІЛЬКИ В ПОВІДОМЛЕННІ

0A>Підказка

0D>Розпізнаний голосом FF>Помилка

Деталі

Перевіртесистемніналаштування (00)

Використовуйте команду "Перевіритисистемніналаштування", щобперевіритипоточніналаштування модуля розпізнавання голосу, включитипослідовнушвидкістьпередачіданих, режим вихідного вводу-виводу, ширину імпульсувихідного вводу-виводу, автоматичнезавантаження та функціюгрупи.

Формат:

| AA | 02 | 00 | 0A |

Повернення:

| AA | 08 | 00 | ДПА | BR | MOM | IOPW | AL | ВРП | 0A |

Табл. 1.1. Таблиця характеристик

	Опис
ДПА	Навченийстатус 0-нетренований 1-навчений Значеннязапису FF виходитьзамежідіапазону
БР	Швидкістьпередачіданих 0 або 3 -9600 1 - 2400 2 - 4800 4 - 19200 5 - 38400

Приклад:

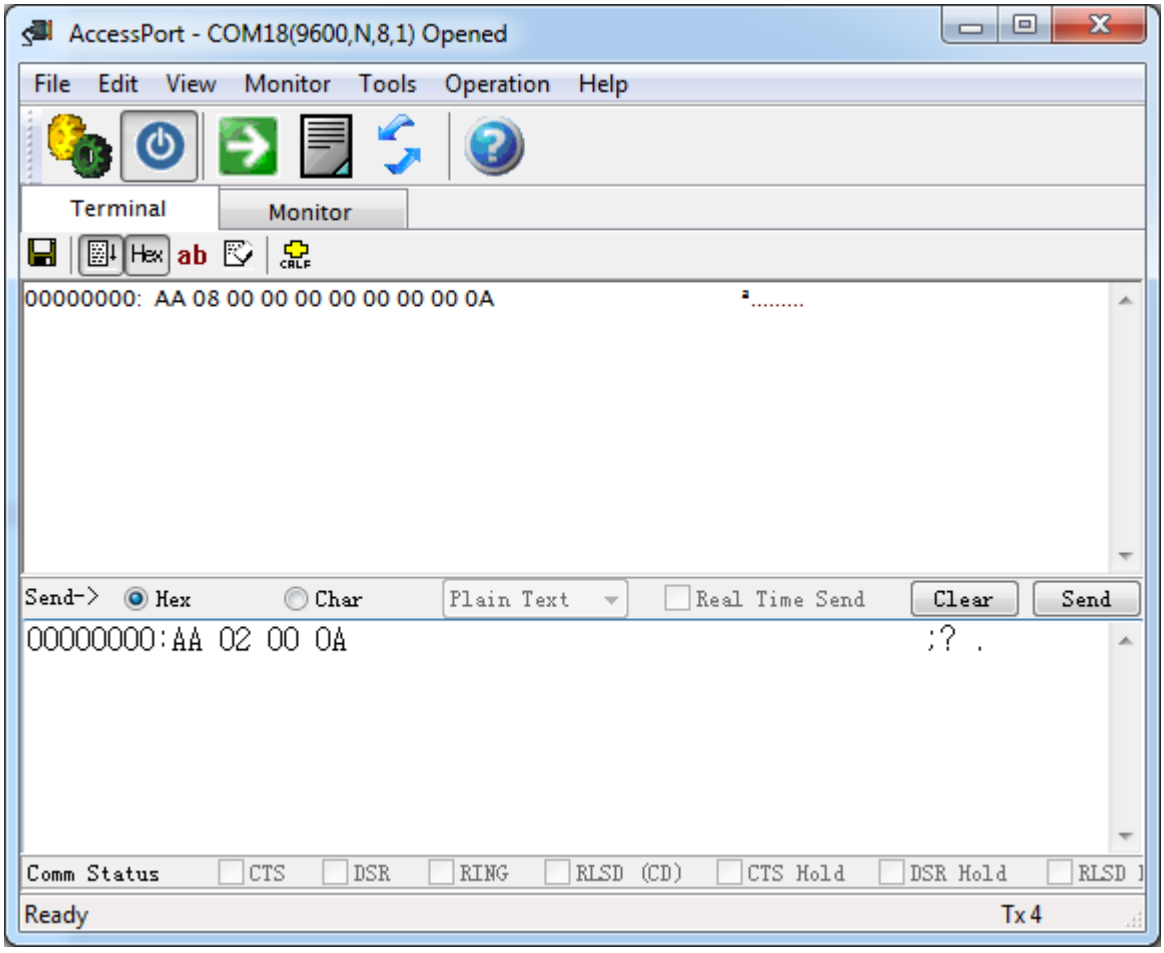

Рис. 1.11. Вікно терміналу

Розпізнаваччеків (01)

Використовуйте команду "Перевіритирозпізнавач", щобперевіритирозпізнавач модуля розпізнавання голосу.

Формат:

| AA | 02 | 01 | 0A |

Повернення:

| AA | 0D | 01 | RVN | VRI0 | VRI1 | VRI2 | VRI3 | VRI4 | VRI5 | VRI6 | RTN | VRMAP | GRPM | 0A |

Табл. 1.2. Список характеристик

	Опис
RVN:	Кількістьдійснихголосовихкоманд у розпізнавачі. МАКС. 7
VRIn	n = 0 ~ 6 Голосовікоманди в розпізнавачі, n - значенняіндексурозпізнавача
RTN	Кількістьзагальнихзаписів у розпізнавачі.
VRMAP	Дійснакартабітівкоманддля VRI0 ~ VRI6.
GRPM	Груповийрежим FF - не в груповомурежимі 00 ~ 0A - системнагрупа 80 ~ 87 - режимгрупикористувачів

Приклад

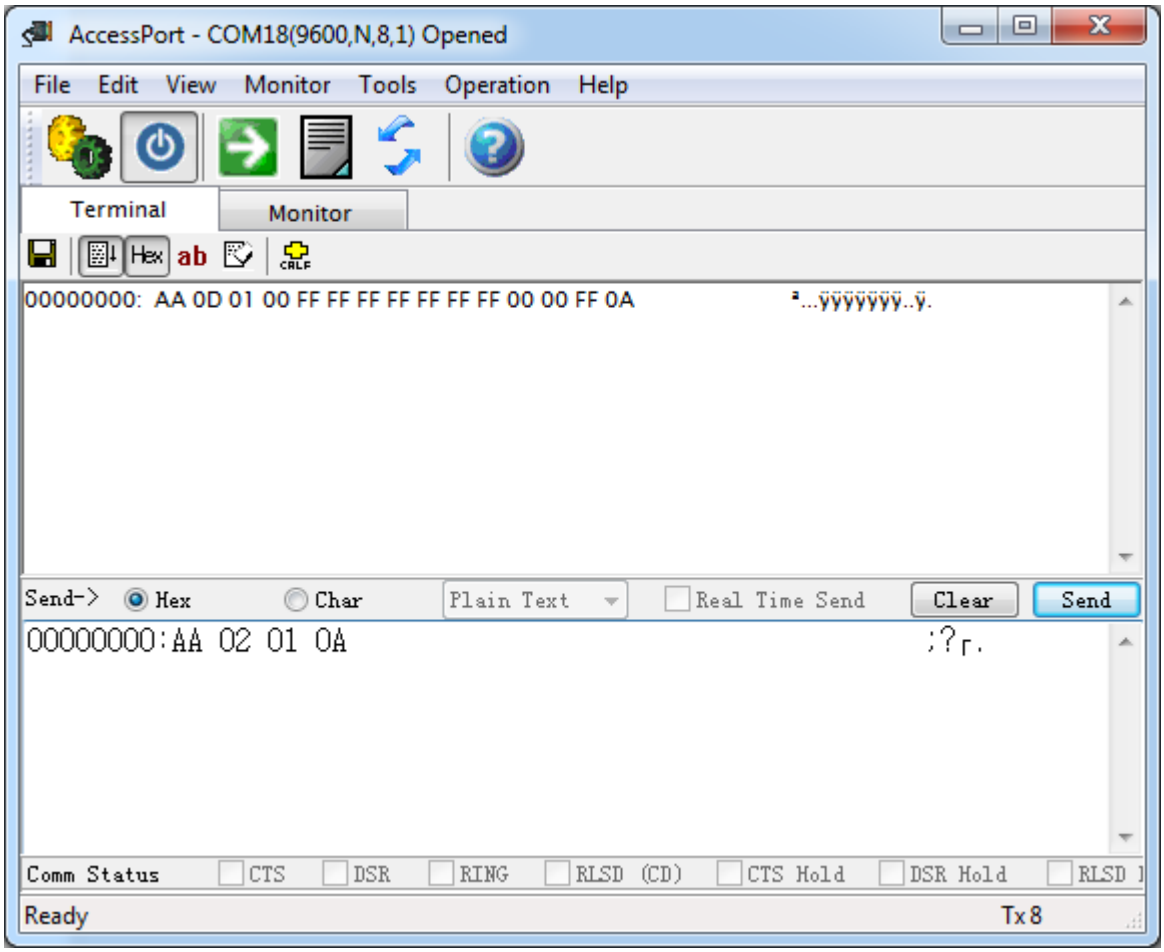

Рис. 1.12. Вікно терміналу

Перевірка стану записупоїзда (02)

Використовуйте команду "Перевірити стан поїздазапису", щобперевірити, читренуєтьсязапис.

Формат:

Перевіртевсі записи

| AA | 03 | 02 | FF | 0A |

Перевіртевказані записи

| AA | 03 + n | 02 | R0 | ... | Rn | 0A |

Повернення:

| AA | 5 + 2 * n | 02 | N | R0 | ДПА | ... | Rn | ДПА | 0A |

Табл. 1.3. Список характеристик

	Опис
N	Кількістьпідготовленихзаписів.
R0 ~ Rn	Записголосу.
ДПА	Навченийстанголосовихкоманд 0 - нетренований 1 - навчений FF - значеннязаписупозадіапазоном

Приклад:

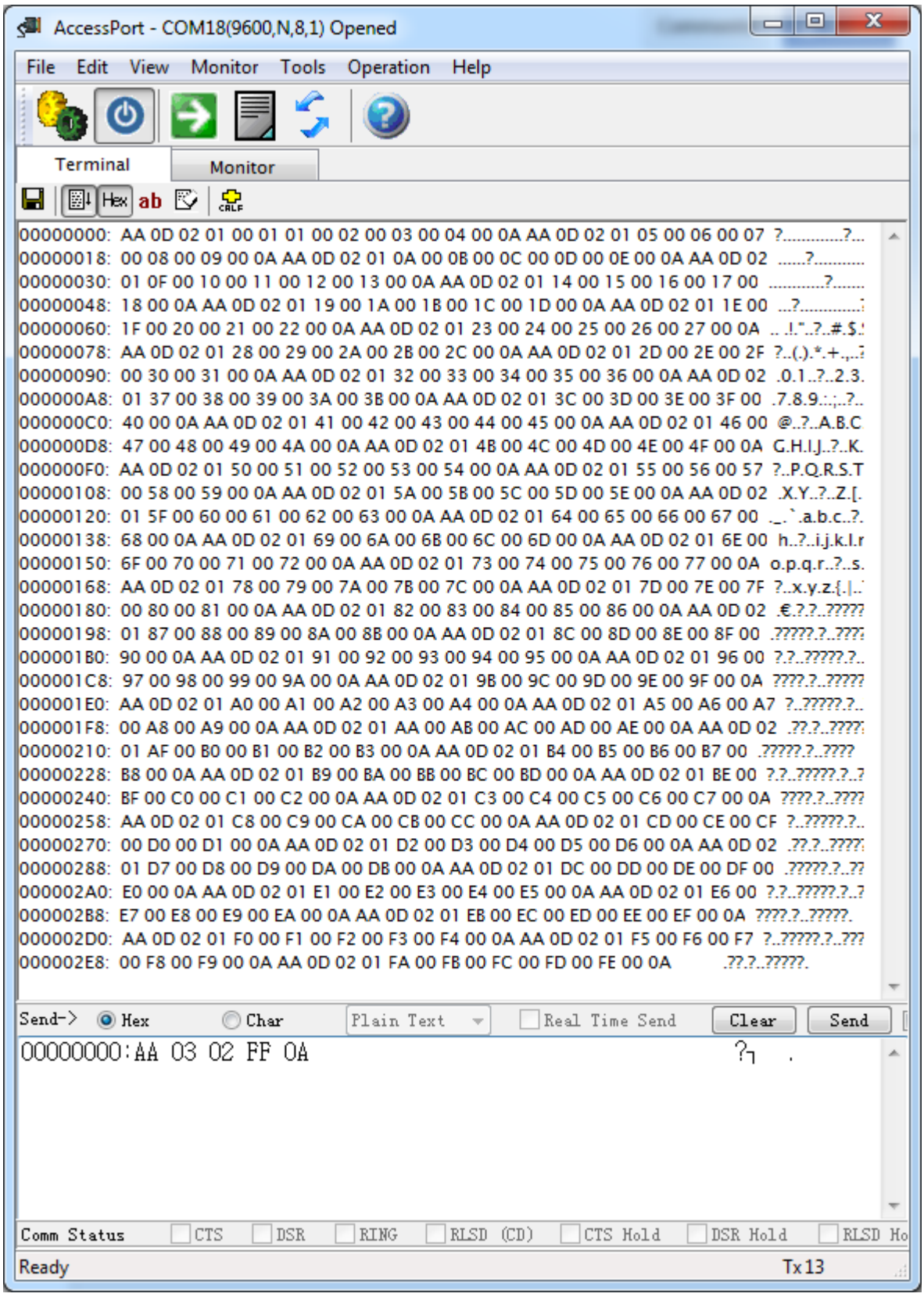

Рис. 1.13. Вікно терміналу

1.5. Проект інтелектуальних інвалідних колясок MIT, щорозробляє робот-інвалідний візок із голосовим керуванням

Бачення: проектуваннярозумнихінвалідних колясок майбутнього

Новіінженернірозробкидаютьможливістьрозробитирозумнідопоміжнітехнології для інваліднихвізків, якіможутьпокращитижиттябагатьох людей, якікористуютьсяінваліднимивізками. У ційроботірозробляються інтелектуальніінваліднівізки завтрашньогодня :розробляється інтелектуальнийінваліднийвізок ізголосовимуправлінням, якийзнає про своєоточення, щобвінмігдопомогтисвоємукористувачеві у різноманітнихзавданнях.

Метою цьогорозумного проекту для інвалідних колясок є вдосконаленнязвичайногоінвалідноговізка з використанням датчиків для сприйняттяоточенняінвалідноговізка , мовногоінтерфейсу для інтерпретації команд, бездротового пристрою для визначеннямісцярозташування на рівнікімнати та програмногозабезпеченняуправліннядвигуном для здійснення руху інвалідноговізка.

Роботизованийінваліднийвізоквивчає структуру навколишньогосередовища (лікарня, реабілітаційний центр, будиноктощо) за допомогоюрозповідноїекскурсії, проведеноїкористувачемабовихователямикористувача. Згодомкрісло-коляска можепереїхатив будь-яке ранішеназванемісце за голосовою командою (наприклад, "Відведи мене до їдальні"). Цятехнологіяпідходить для людей, яківтратилирухливість через травму мозкуабовтратукінцівок, але якізберігаютьмову. Цятехнологіяможетакожпідвищитибезпеку для користувачів, яківикористовуютьзвичайніінвалідні коляски, керовані джойстиком, запобігаючизіткненнюзістінами, нерухомими предметами, меблями та іншими людьми.

Передбачається, щоінваліднекрісло, якимможнакерувати голосом, можепокращитиякістьжиття та безпекудесятківтисячкористувачів. Більше того, значніполіпшення стану здоров'я та

економіякоштівможутьвідбутисязавдякизменшеннюабоусуненню травм, спричиненихзіткненнями, таких як рани та переломи кінцівок. Команда викладачів, студентів та дослідників походить з декількох кафедр (аеронавтики та астронавтики; електротехніки та обчислювальної техніки; підрозділу інженерних систем) та лабораторій (Лабораторія комп'ютерних наук та штучного інтелекту (CSAIL) та MITAgeLab) з усього MIT. Розробкацьогоінтелектуальногокріслаохоплюєдекількадоменів, включаючиробототехніку, штучнийінтелект, машинненавчання, взаємодіюлюдини з комп'ютером та дизайн інтерфейсукористувача, системирозпізнаваннямови та роль технології для людей з обмеженимиможливостями та людей, якістаріють.[3]

1.6. Порівняння існуючих рішень

Згідно з попередніми даними була проведена порівняльна характеристика описаних рішень.

Перевагами Wekinatorє те, що він використовує розрахункові потужності комп’ютера, до якого під’єднаний, це надає можливість використовувати широкий спектр типів нейронних мереж, а також використання великих датасетів та будь-яких алгоритмів оптимізації. Але у цього способу є великий недолік – відсутність автономності. Прилад отримує від комп’ютера вже готовий клас команди, а сама плата лише утворює команди управління. Такий спосіб реалізації використовує мікрофон компютера для отримання команди, та його ресурси для її розпізнавання. Це не дає можливості використовувати данне рішення для обраної для данної дисертації області застосування.

Наступним є рішення з використанням BT VOICE CONTROL для ARDUINO. Данне рішення є більш автономним, адже плата живитсявід аккумулятора, та не потребує фізичного під’єднання до інших пристроїв. Це надає змогу використовувати данний спосіб реалізаціїв приладах, що не повинні бути

обмежені у пересуванні. Але данний союз плати с телефоном вимагає постійного з’єднання через пристрій Bluetooth. Такий спосіб реалізації також надає багате розмаїття алгоритмів розпізнавання. Недоліком є те, що данний підхід використовує застарілі апаратні складові, що унеможливлює повну автономність, необхідну для реалізації проекту, а також вимагає використовувати мікрофон телефону для отримання голосових команд.

На відміну від попередніх рішень ElechouseVoiceRecognitionModule v3.1 отримує повну автономність роботи, що є його найбільшою перевагою. Також перевагою є те, що це вже готовий модуль, що не потребує складних налаштувань та може бути одразу використаний у системах на основі плат Arduino. Недоліками цього рішення є не тільки використання застарілих методів, але й обмежений словник та довжина команд.

Останнім серед представлених рішень є роботизована інвалідна коляска від MIT. Перевагами є те, що вона не тільки автономна, та дозволяє використовувати великий словник, але до того надає можливість отримувати інформацію щодо навколишнього оточення за допомогою додаткових сенсорів. Недоліками цього рішення є те, що така реалізація є дорогою та не підходить для використання в Україні через мовні обмеження.

Нижче представлена таблиця порівняння існуючих рішень.

Табл. 1.4. порівняльна характеристика

Критерій	Wekinator	BT VOICE CONTROL	VoiceRecognitionModule	коляска від MIT
Автономність	-	-	+	+
Сучасні засоби	-	-	-	+
Широкий словник	+	+	-	+
Виокристання української мови	-	-	-	-

Отже необхідно розробити автономну систему, що дозволяла б використовувати її автономно для управління інвалідним візком. Також система має використовувати сучасні засоби реалізації для оптимізації процесу, його прискорення та збільшення точності. Дані вимоги повинні бути виконані в рамках використання українсько-мовних команд.

Можна виділити наступні функції та властивості, якими повинна володіти система розпізнавання команд голосового управління для пристроїв на базі ардуіно:

- Функція розпізнавання 6-ти команд управління українською мовою
- Використання плати як обчислювальої техніки для розпізнавання команд.
- Автономність та відсутність необхідності використання додаткових обислювальних засобів для розпізнавання команд.

Висновокрозділу 1

В даномурозділірозглянутііснуючісистеми розпізнавання командуправління для пристроїв на базі Arduino. Згідно з викладенимматеріаломможназазначити, щовже існують схожі рішення проблеми, але їх головний недолік – загальність, та у більшості випадків, використання допоміжних засобів безпосередньо при роботі системи, що ускладнює маштабованість та компактність системи. Середіншого, деякітакіпрограмивикористовуютьзастаріліметодирішення проблеми. І хочаіснуютьпрограми, щонадаютькористувачузручнийінтерфейс, а такожмають велику кількістьналаштувань, такіпрограминайчастішенадаютьлишечастинуфункціональностіпрограми, необхідної для вирішення поставленої задачі.

РОЗДІЛ 2 МЕТОДИ ТА АЛГОРИТМИ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

2.1. Бібліотека Librosa

Нова наукова область пошуку музичної інформації (MIR) широко охоплює теми на стику музикознавства, цифрової обробки сигналів, машинного навчання, пошуку інформації та бібліотечної справи. Незважаючи на те, що область досить молода - перший міжнародний симпозіум з пошуку музичної інформації (ISMIR) 1 відбувся у жовтні 2000 р., - вона швидко розвивається, частково завдяки поширенню та практичним науковим потребам цифрових музичних послуг, таких як iTunes, Пандора та Spotify. Хоча перевагу досліджень MIR проводили за допомогою спеціальних інструментів та сценаріїв, розроблених дослідниками на різних мовах, таких як MATLAB або C ++, стабільність, масштабованість та простота використання цих інструментів часто залишали бажати кращого.

Останніми роками серед спільноти MIR зріс інтерес до використання (наукового) Python як життєздатної альтернативи. Це обумовлено злиттям кількох факторів, включаючи наявність високоякісних бібліотек машинного навчання, таких як scikit-learn [Pedregosa11] та інструментів, заснованих на Theano [Bergstra11], а також великого каталогу пакетів Python для роботи з текстом дані та веб-служби. Однак прийняття Python сповільнилося відсутністю стабільної базової бібліотеки, яка забезпечує основні процедури, на яких побудовано багато програм MIR. Щоб виправити цю ситуацію, була розробиленalibrosa: пакет Python для обробки звукових та музичних сигналів.

Розробляючи ліброзу, визначилися кілька ключових концепцій. По-перше, прагнення забезпечити низький бар'єр для входу для дослідників, знайомих з MATLAB. Зокрема, обрати відносно плоский макет пакету, і після scipy [Jones01]

покладатись на типи даних і функції numpy [VanDerWalt11], а не на абстрактні ієрархії класів.

По-друге, витрачені значні зусилля на стандартизацію інтерфейсів, імен змінних та (за замовчуванням) параметрів параметрів для різних функцій аналізу. Це завдання ускладнювалося тим, що посилальні реалізації, з яких походять реалізації, походять від різних авторів і часто розробляються як разові сценарії, а не як належні бібліотечні функції з чітко визначеними інтерфейсами.

По-третє, де це можливо, збережено зворотну сумісність із існуючими еталонними реалізаціями. Це досягається за допомогою регресійного тестування на чисельну еквівалентність результатів. Всі тести реалізовані в рамках "nose".

По-четверте, оскільки MIR - це швидко розвивається галузь, точні реалізації, надані librosa, можуть не відображати стан техніки для будь-якого конкретного завдання. Отже, функції розроблені як модульні, що дозволяє практикам надавати свої власні функції, коли це доречно, наприклад, спеціальна оцінка сили початку може бути надана трекеру ритму як аргумент функції. Це дозволяє дослідникам використовувати існуючі бібліотечні функції, експериментуючи з удосконаленнями певних компонентів. Хоча це здається простим і очевидним, з практичної точки зору монолітні конструкції та відсутність взаємодії між різними базами досліджень історично ускладнювали це.

Нарешті, метою є читабельний код, ретельна документація та вичерпне тестування. Вся розробка ведеться на GitHub. Застосовуються сучасні практики розробки програмного забезпечення, такі як постійне тестування інтеграції (через Travis5) та покриття (через Coveralls6). Всі функції реалізовані на чистому Python, ретельно задокументовані за допомогою Sphinx, і включають приклад коду, що демонструє використання. Реалізація здебільшого відповідає рекомендаціям PEP-8, з невеликим набором винятків для імен змінних, які роблять код більш стислим, не жертвуючи ясністю: наприклад, y та sr надаються перевагу більш детальним іменам, таким як audio_buffer та sampling_rate.

2.2. TensorFlowLite

TensorFlowLite - ценабірінструментів, якідопомагаютьрозробникамзапускатимоделіTensorFlow на мобільних, вбудованих та IoT-пристроях. Цедозволяєзробитивисновок машинного навчання на пристрої з низькоюзатримкою та невеликим двійковимрозміром.

TensorFlowLiteскладається з двохосновнихкомпонентів:

ПерекладачTensorFlowLite , якийпрацюєспеціальнооптимізованімоделі на багатьохрізнихтипівобладнання, в тому числімобільнихтелефонів, вбудованихпристроївLinux і мікроконтролерах.

Конвертер TensorFlowLite , якийперетворює модель TensorFlow в ефективну форму для використанняінтерпретатора, іможе ввести оптимізацію, щобполіпшитибінарнийрозмір і продуктивність.

Машинненавчання на краю

TensorFlowLiteрозроблений, щобспроститимашинненавчання на пристроях, "на краю" мережі, замість того, щобнадсилатидані вперед і назад із сервера. Для розробниківвиконання машинного навчання на пристроїможедопомогтипокращити:

Затримка: немаєзворотного шляху до сервера

Конфіденційність: дані не повиннізалишатипристрій

Підключення: підключення до Інтернету не потрібно

Споживанняенергії: мережевіз'єднанняпотребуютьенергії

TensorFlowLiteпрацює з величезним набором пристроїв, відкрихітнихмікроконтролерів до потужнихмобільнихтелефонів.

Ключовіособливості

Інтерпретатор, налаштований на вбудовануML ,підтримуєнабіросновнихоператорів, оптимізованих для вбудованихпрограм, і має невеликий двійковийрозмір.

Різнапідтримкаплатформи , щоохоплює пристрої Android та iOS , вбудованийLinux та мікроконтролери, використовуючи API платформи для прискореноговиведення.

API для декількохмов, включаючиJava, Swift, Objective-C, C ++ та Python.

Високапродуктивність , апаратнеприскорення на підтримуванихпристроях, оптимізовані для пристроїв ядра та попередньосплавленіактивації та упередження .

Засобиоптимізаціїмоделей ,включаючи квантування , якіможутьзменшитирозмір та підвищитипродуктивність моделей без шкоди для точності.

Ефективний формат моделі ,використовуючи FlatBuffer, якийоптимізований для невеликих розмірів та портативності.

Попередньонавченімоделі для загальнихзавдань машинного навчання, якіможнаналаштувативідповідно до вашоїпрограми.

Зразки та навчальніпосібники, якіпоказують, як розгорнутимоделі машинного навчання на підтримуваних платформах.

Технічніобмеження

TensorFlowLiteплануєзабезпечитивисокоефективнийвисновок на пристрої для будь-якоїмоделіTensorFlow. ОднакінтерпретаторTensorFlowLite в даний час підтримуєобмеженупідмножинуоператорівTensorFlow, якіоптимізовані для використання на пристрої. Цеозначає, щодеякімоделівимагаютьдодатковихкроків для роботи з TensorFlowLite.

Щобдізнатись, якіоператоридоступні, див. Сумісністьоператорів .

Якщо у вашіймоделівикористовуютьсяоператори, якіще не підтримуютьсяінтерпретаторомTensorFlowLite, ви можете використовувати TensorFlowSelect, щобвключитиопераціїTensorFlow до вашоїзбіркиTensorFlowLite. Однакцепризведе до збільшеннядвійковогорозміру.

НаразіTensorFlowLite не підтримуєнавчання на пристрої, але воно є в нашій Дорожнійкарті , поряд з іншимизапланованимивдосконаленнями.[4]

2.2. Keras та послідовна модель

Keras -

це бібліотека з відкритимкодом, якаозабезпечуєінтерфейсPythonдля штучнихнейро нних мереж. Kerasдіє як інтерфейс для бібліотеки TensorFlow.

До версії 2.3 Kerasпідтримувавдекількасерверних систем, включаючи TensorFlow, MicrosoftCognitiveToolkit , R , Theano та PlaidML. Створений для швидкогоекспериментування з глибокиминейронними мережами , вінфокусується на тому, щоб бути зручним, модульним та розширюваним. Вінбуврозроблений в рамках дослідницькихробіт проекту ONEIROS (Відкрита нейро-електроннаінтелектуальнаопераційна система роботів) а йогоголовним автором та супровідником є Франсуа Шолле, інженер Google . Шоллеттакож є автором моделіглибокихнейронних мереж XCeption.

Особливості:

Kerasміститьчисленніреалізаціїзагальновживанихнейромережевихбудівельнихбло ків, таких як шари, цілі, функціїактивації, оптимізатори та безлічінструментів для спрощенняроботи з данимизображень та тексту для спрощеннякодування, необхідного для написанняглибокого коду нейронноїмережі. Код розміщений на GitHub, а форумипідтримкиспільнотивключаютьсторінку з проблемами GitHub та канал Slack .

На додаток до стандартнихнейронних мереж, Kerasпідтримує згорткові та повторюванінейроннімережі . Вінпідтримуєіншізагал ьноприйнятірівніутиліти, такі як випадання, пакетнанормалізація та об'єднання .

Keras дозволяє користувачам виробляти глибокі моделі на смартфонах (iOS та Android), в Інтернеті або на віртуальній машині Java .Цетакождозволяєвикористовуватирозподілененавчання моделей глибокогонавчання на кластерах графічнихпроцесорів (GPU) і тензорнихблоків (TPU)[5]

KerasSequential

Вказівка вхідноїформи

Модель повинна знати, яку форму введення вона повинна очікувати. З цієї причини перший шар у `Sequential` моделі (і лише перший, оскількинаступнішариможутьробитиавтоматичневиведенняфігури) повинен отримуватиінформацію про свою вхідну форму. Є кількаможливихспособівзробитице:

Можна передати `input_shape` аргументпершому шару. Це кортеж фігури (кортеж цілихчи `None` записів, де `None` вказує на те, щоможнаочікувати будь-яке додатнеціле число). У `input_shape`, розмірпартії не входить.

Можна передайтизамістьцього `batch_input_shape` аргумент, де включений розмірпартії. Цекорисно для вказівкифіксованогорозмірупартії (наприклад, із RNN, щомістятьдані про стан).

Деякі 2D-шари, такі як `Dense`, підтримуютьспецифікаціюсвоєївхідноїформи через аргумент `input_dim`, а деякі 3D-часові шарипідтримуютьаргументи `input_dim` та `input_length`.

Таким чином, наступні три фрагментисувороеквівалентні:

```
model = Sequential()
```

```
model.add(Dense(32, input_shape=(784,)))
```

```
model = Sequential()
```

```
model.add(Dense(32, batch_input_shape=(None, 784)))
```

```
# notethatbatchdimensionis "None" here,
```

```
# sothemodelwillbeabletoprocessbatchesofanysize.
```

```
model = Sequential()
```

```
model.add(Dense(32, input_dim=784))
```

А такожтакі три фрагменти:

```
model = Sequential()
```

```
model.add(LSTM(32, input_shape=(10, 64)))
```

```
model = Sequential()
```

```
model.add(LSTM(32, batch_input_shape=(None, 10, 64)))
```

```
model = Sequential()
```

```
model.add(LSTM(32, input_length=10, input_dim=64))
```

Шар злиття

Кілька `Sequential` екземплярівможнаоб’єднати в один вихід через `Merge` шар. Вихід - це шар, якийможнадодати як перший шар у новій `Sequential` моделі. Наприклад, ось модель іздвомаокремимигілками вводу, якіоб’єднуються:

```
from keras.layers import Merge
```

```
left_branch = Sequential()
```

```
left_branch.add(Dense(32, input_dim=784))
```

```
right_branch = Sequential()
```

```
right_branch.add(Dense(32, input_dim=784))
```

```
merged = Merge([left_branch, right_branch], mode='concat')
```

```
final_model = Sequential()
```

```
final_model.add(merged)
```

```
final_model.add(Dense(10, activation='softmax'))
```

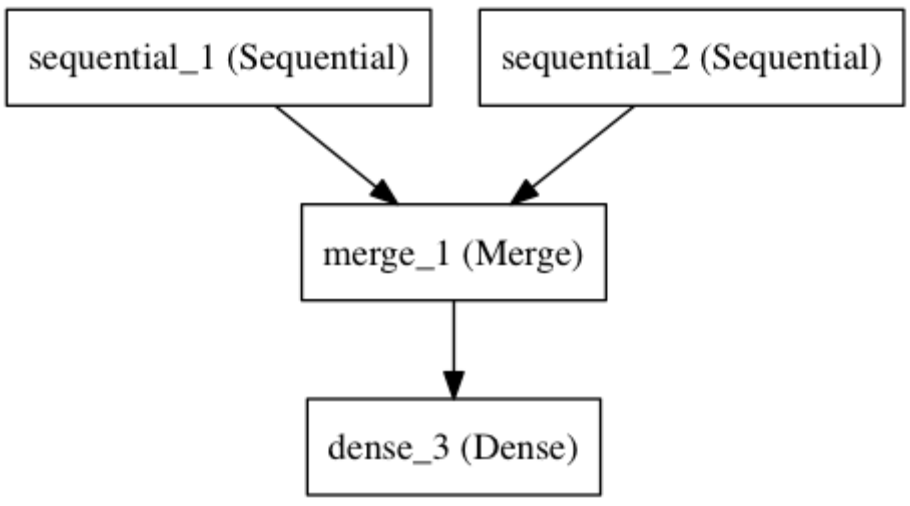

Рис. 2.1. Діаграма моделі

Потімтакудвогалузеву модель можнанавчити, наприклад:

```
final_model.compile(optimizer='rmsprop', loss='categorical_crossentropy')
```

```
final_model.fit([input_data_1, input_data_2], targets) #
```

wepassonedataarraypermodelinput

Merge Шарпідтримуєкільказдалегідьвизначенихрежимів:

sum (за замовчуванням): сума за елементами

concat: конкатенація тензора. Ви можете вказативіськонкатенації за допомогою аргументу concat_axis.

mul: елементнемноження

ave: тензор середній

dot: крапковийвиріб. Ви можете вказати, якіосізменшувати за допомогою аргументу dot_axes.

cos: близькістькосинусівміж векторами у 2D тензорах.

Такожможнапередатифункцію як mode аргумент, дозволяючидовільніперетворення:

```
merged = Merge([left_branch, right_branch], mode=lambda x: x[0] - x[1])
```

Компіляція

Перш ніжнавчати модель, потрібноналаштуватипроцеснавчання, якийвиконується за допомогою compile методу. Вінотримує три аргументи:

оптимізатор. Цеможе бути ідентифікатор рядка існуючогооптимізатора (наприклад, `rmsprop` or `adagrad`)

абоекземпляр `Optimizer` класу. Див .: оптимізатори .

функціявтрат. Це мета, яку модель намагатиметьсязвести до мінімуму. Цеможе бути ідентифікатор рядка існуючоїфункціївтрат (наприклад, `categorical_crossentropy` або `mse`), а може бути цільовоюфункцією. Див .: цілі .

список метрик. Для будь-якоїпроблемикласифікаціїпотрібно буде встановитицезначення `metrics=['accuracy']`. Метрика може бути ідентифікатором рядка існуючої метрики абокористувацькоїфункції метрики. Спеціальнаметричнафункція повинна повертатиабооднезначення тензора, абодікт `metric_name ->metric_value`. Див .: метрики .

```
# for a multi-classclassificationproblem

model.compile(optimizer='rmsprop',

loss='categorical_crossentropy',

metrics=['accuracy'])


# for a binaryclassificationproblem

model.compile(optimizer='rmsprop',

loss='binary_crossentropy',
```

```
metrics=['accuracy'])

# for a meansquarederrorregressionproblem
model.compile(optimizer='rmsprop',
              loss='mse')

# forcustommetrics
import keras.backend as K

def mean_pred(y_true, y_pred):
    return K.mean(y_pred)

def false_rates(y_true, y_pred):
    false_neg = ...
    false_pos = ...
```

```
return {
'false_neg': false_neg,
'false_pos': false_pos,
    }
model.compile(optimizer='rmsprop',
loss='binary_crossentropy',
metrics=['accuracy', mean_pred, false_rates])
```

Навчання

МоделіKerasнавчаються на масивахNumpyвхіднихданих та міток. Для навчаннямоделізазвичайвикористовується `fit` функція.

```
# for a single-inputmodelwith 2 classes (binary):
model = Sequential()
model.add(Dense(1, input_dim=784, activation='sigmoid'))
model.compile(optimizer='rmsprop',
```

```
loss='binary_crossentropy',

metrics=['accuracy'])


# generatedummydata

importnumpyasnp

data = np.random.random((1000, 784))

labels = np.random.randint(2, size=(1000, 1))

# trainthemodel, iteratingonthedatainbatches

# of 32 samples

model.fit(data, labels, nb_epoch=10, batch_size=32)

# for a multi-inputmodelwith 10 classes:

left_branch = Sequential()

left_branch.add(Dense(32, input_dim=784))

right_branch = Sequential()
```

```
right_branch.add(Dense(32, input_dim=784))

merged = Merge([left_branch, right_branch], mode='concat')

model = Sequential()

model.add(merged)

model.add(Dense(10, activation='softmax'))

model.compile(optimizer='rmsprop',

loss='categorical_crossentropy',

metrics=['accuracy'])

# generatedummydata

importnumpyasnp

fromkeras.utils.np_utilsimportto_categorical

data_1 = np.random.random((1000, 784))

data_2 = np.random.random((1000, 784))
```

```
# theseareintegersbetween 0 and 9
labels = np.random.randint(10, size=(1000, 1))
# weconvertthelabelsto a binarymatrixofsize (1000, 10)
# forusewithcategorical_crossentropy
labels = to_categorical(labels, 10)
# trainthemodel
# notethatwearepassing a listofNumpyarraysastrainingdata
# sincethemodelhas 2 inputs
model.fit([data_1, data_2], labels, nb_epoch=10, batch_size=32)
```

2.3. Моделі классифікації

2.3.1. Лінійна класифікація

Основна ідея лінійного класифікатора полягає в тому, що простору ознак може бути розділене гиперплоскостью на дві півплощини, в кожній з яких прогнозується одне з двох значень цільового класу. Якщо це можна зробити без помилок, то навчальна вибірка називається лінійно нероздільні.[6]

2.3.2. Машина опорних векторів

Основна ідея методу - переклад вихідних векторів в простір більш високої розмірності і пошук розділяє гіперплоскості з максимальним зазором в цьому просторі. Дві паралельні гіперплощини будуються по обидва боки гіперплощини, що розділяє класи. Розділяючагіперплоскість максимізує відстань до двох паралельних гіперплоскостей. Алгоритм працює в припущенні, що чим більша різниця або відстань між цими паралельними гіперплоскостями, тим менше буде середня помилка класифікатора.

На практиці випадки, коли дані можна розділити гиперплоскостью, досить рідкісні. В цьому випадку поступають так: всі елементи навчальної вибірки вкладаються в простір X більш високої розмірності, так, щоб вибірка була лінійно роздільна.[7]

2.3.3. Дерева рішень

Дерева рішень використовуються в повсякденному житті в самих різних областях людської діяльності.

До впровадження масштабованих алгоритмів машинного навчання в банківській сфері завдання кредитного скорингу вирішувалася експертами. Рішення про видачу кредиту позичальникові приймалося на основі деяких інтуїтивно (або з досвіду) виведених правил, які можна представити у вигляді дерева.

В основі популярних алгоритмів побудови дерева рішень лежить принцип жадібної максимізації приросту інформації - на кожному кроці вибирається той ознака, при поділі за яким приріст інформації виявляється найбільшим. Далі процедура повторюється рекурсивно, поки ентропія не опиниться рівною нулю або якийсь малій величині (якщо дерево не підганяється ідеально під навчальну вибірку щоб уникнути перенавчання).[8] плюси:

- Породження чітких правил класифікації, зрозумілих людині, наприклад, "якщо вік <25 та інтерес до мотоциклів, то відмовити в кредиті". Це властивість називають інтерпретованих моделі;
- Дерева рішень можуть легко візуалізувати, як сама модель (дерево), так і прогноз для окремого взятого тестового об'єкта (шлях в дереві);
- Швидкі процеси навчання і прогнозування;
- Мале число параметрів моделі;
- Підтримка і числових, і категоріальних ознак.

мінуси:

- У породження чітких правил класифікації є й інша сторона: дерева дуже чутливі до шумів у вхідних даних, вся модель може кардинально змінитися, якщо трохи зміниться навчальна вибірка (наприклад, якщо прибрати один з ознак або додати кілька об'єктів), тому і правила класифікації можуть сильно змінюватися, що погіршує интерпретируемость моделі;
- Розділяє межа, побудована деревом рішень, має свої обмеження (складається з гіперплоскостей, перпендикулярних якийсь із координатної осі), і на практиці дерево рішень за якістю класифікації поступається деяким іншим методам;

2.3.4. Метод найближчих сусідів

Метод найближчих сусідів (k NearestNeighbors, або kNN) - теж дуже популярний метод класифікації, також іноді використовується в задачах регресії. Це, нарівні з деревом рішень, один з найбільш зрозумілих підходів до класифікації. На рівні інтуїції суть методу така: подивися на сусідів, які переважають, такий і ти. Формально основою методу є гіпотеза компактності: якщо метрика відстані між прикладами введена досить вдало, то схожі приклади набагато частіше лежать в одному класі, ніж в різних.

Для класифікації кожного з об'єктів тестової вибірки необхідно послідовно виконати наступні операції:

Обчислити відстань до кожного з об'єктів навчальної вибірки

Відібрати k об'єктів навчальної вибірки, відстань до яких мінімально

Клас об'єкта - це клас, який найчастіше трапляється серед k найближчих сусідів

Під задачу регресії метод адаптується досить легко - на 3 кроці повертається не мітка, а число - середнє (або медіанне) значення цільового показника серед сусідів.

Примітна властивість такого підходу - його ліниво. Це означає, що обчислення починаються тільки в момент класифікації тестового прикладу, а заздалегідь, тільки при наявності навчальних прикладів, ніяка модель не будується. У цьому відмінність, наприклад, від раніше розглянутого дерева рішень, де спочатку на основі навчальної вибірки будується дерево, а потім відносно швидко відбувається класифікація тестових прикладів.[9]

Якість класифікації / регресії методом найближчих сусідів залежить від декількох параметрів :

- число сусідів
- метрика відстані між об'єктами (часто використовуються метрика Хеммінга, евклідова відстань, косинусное відстань і відстань Маньківського). Відзначимо, що при використанні більшості метрик значення ознак треба масштабувати. Умовно кажучи, щоб ознака "Зарплата" з діапазоном значень до 100 тисяч не вносив більший вклад в відстань, ніж "Вік" зі значеннями до 100.
- ваги сусідів (сусіди тестового прикладу можуть входити з різними вагами, наприклад, чим далі приклад, тим з меншим коефіцієнтом враховується його "голос")

Плюси і мінуси методу найближчих сусідів

плюси:

- Проста реалізація;
- Непогано вивчений теоретично;
- Як правило, метод хороший для першого рішення задачі;
- Можна адаптувати під потрібне завдання вибором метрики або ядра (ядро може задавати операцію подібності для складних об'єктів типу графів, а сам підхід kNN залишається тим же);
- Непогана інтерпретація, можна пояснити, чому тестовий приклад був класифікований саме так.

мінуси:

- Метод вважається найшвидшим у порівнянні, наприклад, з композиціями алгоритмів, але в реальних задачах, як правило, число сусідів, які використовуються для класифікації, буде великим (100-150), і в такому випадку алгоритм буде працювати не так швидко, як дерево рішень;
- Якщо в наборі даних багато ознак, то важко підібрати відповідні ваги і визначити, які ознаки не важливі для класифікації / регресії;
- Залежність від обраної метрики відстані між прикладами. Вибір за замовчуванням евклідової відстані найчастіше нічим не обгрунтований. Можна відшукати хороше рішення перебором параметрів, але для великого набору даних це забирає багато часу;
- Немає теоретичних підстав вибору певного числа сусідів - тільки перебір (втім, найчастіше це вірно для всіх гіперпараметров всіх моделей). У разі малого числа сусідів метод чутливий до викидів, тобто схильний перенавчатися;

- Як правило, погано працює, коли ознак багато, через "прокляття розмірності". Про це добре розповідає відомий в ML-співтоваристві професор PedroDomingos - тут в популярній статті "A FewUsefulThingstoKnowaboutMachineLearning", також "thecurseofdimensionality" описується в книзі DeepLearning в розділі "MachineLearningbasics".

2.3.5. Обгрунтування вибору Лінійної Классифікації

Для використання в умоваї обмежених апаратних ресурсів необхідно вибрати просту та точну модель класифікації. Під ці параметри підходить лінійна модель. Також доатковою умовою є те, що апаратна плата, що буде використовуватись у проєкті, підтримує лише бібліотеку Tensorflow Lite. Беручи до уваги ці фактори лінійна регресія є найкращим вибором, бо задовільняє необхідні умови, поставлені до вибору класифікатора.

2.4. Алгоритми оптимізації

2.4.1. NesterovAcceleratedGradient

Цей метод є методом з накопиченням імпульсу. Сама по собі ідея методів з накопиченням імпульсу до очевидності проста: «Якщо ми деякий час рухаємося в певному напрямку, то, ймовірно, нам слід туди рухатися деякий час і в майбутньому». Для цього потрібно вміти звертатися до недавньої історії змін кожного параметра. Можна зберігати останні n примірників Delta на кожному кроці по-чесному вважати середнє, але такий підхід займає надто багато пам'яті для великих n. На щастя, не потрібно точне середнє, а лише оцінка, тому можна скористатися експоненціальним ковзним середнім:

Щоб накопичити що-небудь, будемо множити вже накопичене значення на коефіцієнт збереження $0<\gamma<1$ і додавати чергову величину, помножену на $1-\gamma$. Чимближчеγдоодиниці, тимбільшевікнонакопичення і сильнішезгладжування-

історіяхпочинаєвпливатисильніше, ніжкожнечерговех. Якщох = 0зякогосьмоменту, v_tзгасають у геометричнійпрогресії, експоненціально, звідси і назва. Застосуємоекспоненціальнесереднє, щобнакопичуватиградієнтцільовоїфункціїнашоїмережі:

Де γзазвичай береться як 0.9. γне пропало, а включилося в θ; іноді можна зустріти і варіант формули з явним множником. Чим менше γ, тим більше алгоритм поводиться як звичайний SGD. Якщо в моментtпідкулькоюбувненульовийухил ( ), а потім він потрапив на плато, він все одно продовжить котитися по цьому плато. Більш того, кулька продовжить рухатися пару оновлень в ту ж сторону, навіть якщо ухил змінився на протилежний. Проте, на кульку діє в'язке тертя і кожну секунду він втрачає 1-γсвоєї швидкості.[10]

2.4.2. Adagrad

Деякі ознаки можуть бути вкрай інформативними, але зустрічатися рідко. Екзотична високооплачувана професія, химерне слово в спам-базі - вони запросто потонути в шумі всіх інших оновлень. Мова йде не тільки про рідкі вхідні параметри. Скажімо, цілком можуть зустрітися рідкісні графічні візерунки, які і в ознаках перетворюються тільки після проходження через кілька шарів згортаючої мережі. Потрібнооновлюватипараметри з оглядкоюнате, наскількитипові ознакивонифіксують. Досягтицьогонескладно: необхіднозберігатидлякожногопараметрамережісумуквадратівйогооновлень. Вонабудевиступати в якостіпроксідлятиповості: якщопараметрналежитьланцюжку нейронів, щочастоактивуються, йогопостійносмикаютьтуди-сюди, а значитьсумашвидконакопичується. Перепишемоформулупоновленняосьтак:

Де - сума квадратів оновлень, а є - згладжує параметр, необхідний, щоб уникнути поділу на 0. У часто обновлялись в минулому параметра велика , значить великий знаменник. Параметр змінився всього раз чи два оновиться в повну силу. є берутьпорядку10^{-6}або 10^{-8}длязовсімагресивногопоновлення, але, яквидно з графіків, цеграєрольтількинапочатку, ближчедосередининавчанняпочинаєпереважувати :

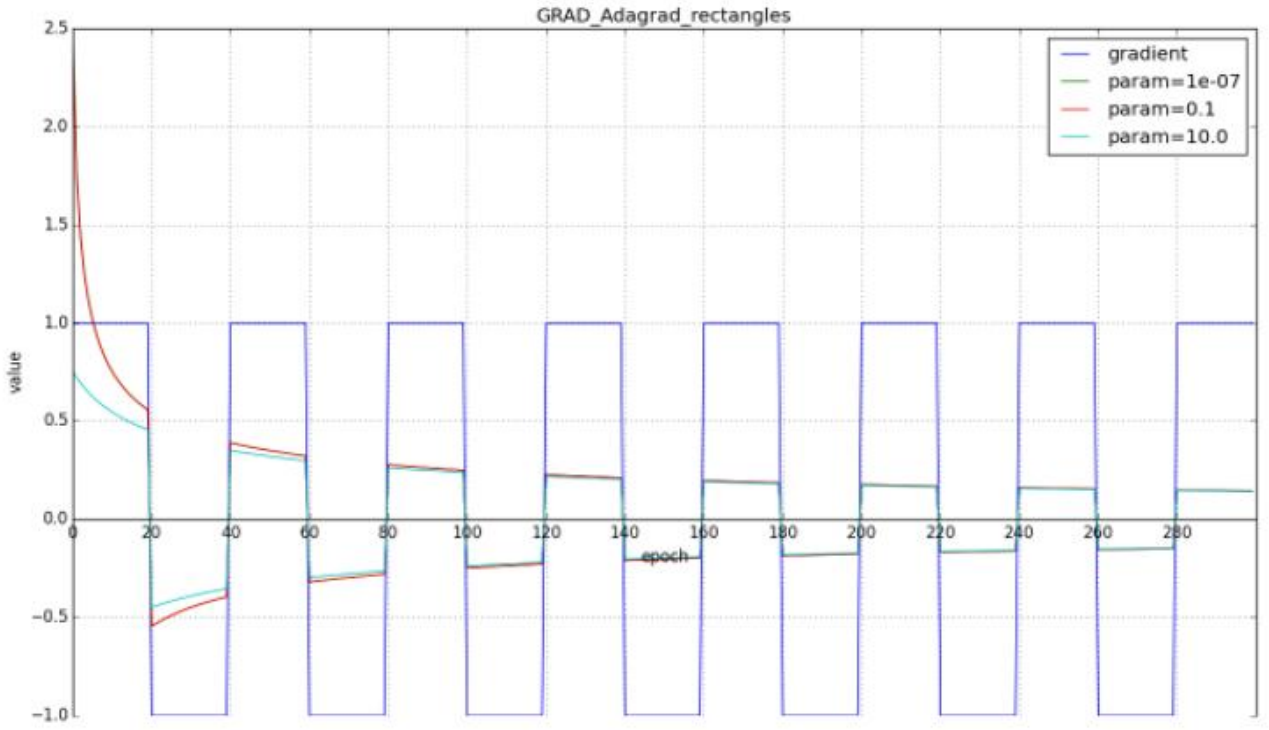

Рис. 2.2. Графік навчання

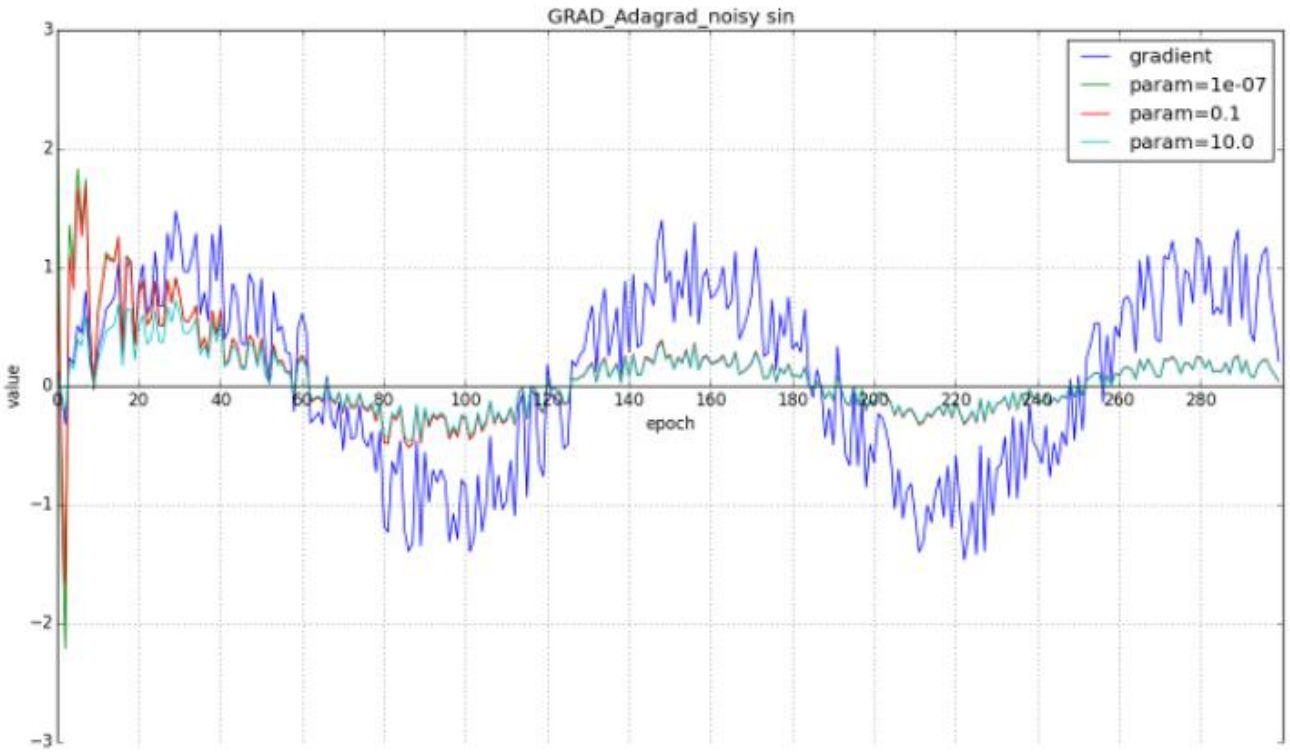

Рис. 2.3. Графік начання на іншій вибірці

2.4.3. Adam

Адам - це алгоритм оптимізації, якийможнавикористовуватизамістьклasичноїпроцедури стохастичного градієнтного спуску для ітеративногопоновлення ваг мережі на основінавчальнихданих.

Адам був представлений ДідерикКінгмен відOpenAI і Джиммі Ба з Університету Торонто в їх 2015 ICLR папір (плакат) підназвою « Адам: метод стохастичною оптимізації «. Я процитуюдокладно з їхстатті в цьомупості, якщо не вказаноінше.

Алгоритм називається Адам. Це не абревіатура і не пишеться як «АДАМ».

... ім'я Адам отримано з адаптивною оцінки моменту.

Представляючи алгоритм, авторипереraховуютьпривабливіперевагивикористання Adam в неопуклих задачах оптимізаціїнаступним чином:

Просто для реалізації.

Обчислювальноефективний.

Маленьківимоги до пам'яті.

Інваріант до зміни масштабу градієнта по діагоналі.

Добре підходить для задач, які є великими з точки зоруданих і / абопараметрів.

Підходить для нестаціонарнихцілей.

Підходить для задач з дужегучними / аборідкіснимиградієнтами.

Гіперпараметри маютьінтуїтивнозрозумілуінтерпретацію і зазвичайвимагаютьневеликої настройки.

Адам відрізняєтьсявідкласичного стохастичного градієнтного спуску.

Стохастичнийградієнтний спуск підтримуєєдинушвидкістьнавчання (звану альфа) для всіхоновлень ваги, і швидкістьнавчання не змінюєтьсяпід час тренування.

Швидкістьнавчанняпідтримується для кожного ваги мережі (параметра) і окремоадаптується в міруrозвиткунавчання.

Метод обчислюєіндивідуальніадаптивнішвидкостінавчання для різнихпараметрів з оцінокпершого і другого моментівградієнтів.

Авториописують Адама як об'єднанняперевагдвохіншихрозширень стохастичного градієнтного спуску. Зокрема:

АдаптивнийГрадієнтний Алгоритм (AdaGrad), якийпідтримуєшвидкістьнавчання по параметру, яка покращуєпродуктивність при проблемах з розрідженимиградієнтами (наприклад, проблеми з природною мовою і комп'ютернимзором).

Середньоквадратичнепоширення (RMSProp), якийтакожпідтримуєшвидкостінавчання по кожному параметру, якіадаптовані на основісередньогозначенняостанніх величин градієнтів для ваги (наприклад, як швидковонозмінюється). Цеозначає, що алгоритм добре справляється з онлайн і нестаціонарнимизавданнями (наприклад, з шумом).

Адам усвідомлюєпереваги як AdaGrad, так і RMSProp.

Замість того щобадаптуватишвидкостінавчанняпараметрів на основісередньогопершого моменту (середнього), як в RMSProp, Адам такожвикористовуєсереднєзначення других моментівградієнтів (нецентрованогодисперсія).

Зокрема, алгоритм обчислюєекспонентнуковзаючусереднюградієнта і квадрата градієнта, а параметри бета1 і бета2 керуютьшвидкістюзагасанняцихковзнихсередніх.

Початковезначеннязміннихсередніх і значень бета1 і бета2, близьких до 1,0 (рекомендується), призводить до зміщенняоцінокмоментів в сторону нуля. Цейзсувдолаєтьсяспочаткуобчисленнямзміщенихоцінок, а потімобчисленнямоцінок з поправкою на зміщення.[11]

2.4.4. Обгрунтування вибору оптимізатора Adam

Як і у випадку з методом классифікації, необходно щоб алгоритм був досить простий, але при цьому швидкий та точний.

Враховуючи характеристики описаних алгоритмів оптимізації вибором став алгоритм Adam, адже він нетільки відповідає на поставлені вимоги, але і на відміну від деяких алгоритмів здатний обходити локальні мінімуми та знаходити оптимальне значення.

Висновки до розділу

У розділі розглянуто різні методи класифікації та алгоритми оптимізації. Докладно розглянуто роботу методів та алгоритмів, показані їх недоліки та переваги. Для кожної зі складової алгоритму розпізнавання команд управління було вказано який алгоритм найбільше підходить для вирішення поставленої задачі.

Виявлено, що для моделі нейронної мережі накраще підходить один з найпростіших алгоритмів. Використання вказаних алгоритмів зумовлено в основному використанням обмежених апаратних ресурсів.

РОЗДІЛ 3. РОЗРОБКА ЗАГАЛЬНОГО АЛГОРИТМУ ДЛЯ ВИРІШЕННЯ ПОСТАВЛЕНОЇ ЗАДАЧІ

3.1. Схема загального алгоритму

Перед тим, як пристрій почне працювати, нейронну мережу необхідно заздалегіть навчити використовуючи власну вибірку, що буде створена індивідуально для користувача для покращення характеристик точності.

Під час навчання система отримує вибірку з набором команд для навчання та тестів, після чого вони подаються до блоків MFCC, які у свою чергурокладаютьїх на мел-частотні характеристики. Після чого ці данні отримує нейронна мережа, покращена за допомогою алгоритма оптимізації.

Піcл того, як нейронна мережа буде навчана, вона конвертується у бібліотеку для середовища розробки на мові Arduino. Ця бібліотека підключається до проекту, разом з яким буде завантажена до плати для виконання задачі розпізнавання команд.

3.2. Мел-частотнікепстральні коефіцієнти

Сучасні системи розпізнавання мови як правило мають ієрархічну модульну структуру. На першому рівні виконується попередня обробка та

виділення акустичних ознак, які характеризують голосову команду. Одним

з найуживаніших на сьогодні методів — є виділення мел-частотних кепстральних коефіцієнтів (Mel-FrequencyCepstralCoefficients або MFCC).

Мел — психофізична одиниця висоти звуку, що пов’язана з частотою за

формулою (3). Отримані на основі цього методу ознаки володіють рядом корисних властивостей — вони легко розраховуються, дають компактне

представлення голосової команди, стійкі до шумових завад з навколишнього середовища.

Наступний рівень систем розпізнавання голосових команд — лінгвістичний. В нього входить процедура пошуку вимовленої команди по словникам еталонів. При розпізнаванні окремих голосових команд, диктор вимовляє слово без оточуючого контексту. Навчання таких систем є трудомісткою задачею і для підвищення надійності зазвичай використовують великі навчальні вибірки (від 5 та більше варіантів вимови однієї голосовоїкоманди). Кожна команда записується в словник еталонів як набір мелчастотнихкепстральних коефіцієнтів. Типова структура такої системи наведена на рис. 1.

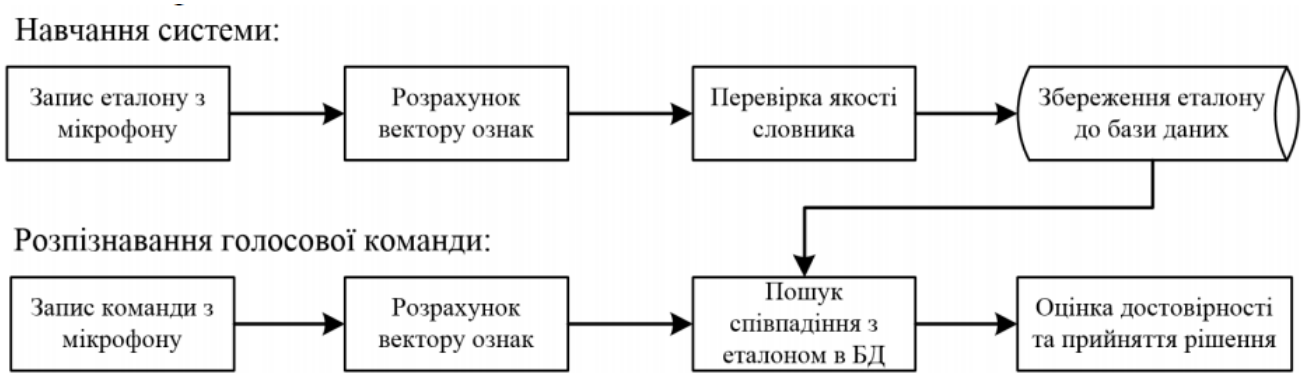

Рис. 3.1. Структура системи

Алгоритм розрахунку мел-частотнихкепстральнихкоефіцієнтів Такий метод отриманняознак є одним з найпоширеніших як в системах розпізнаваннядикторів так і в системах розпізнаваннямови. В вхід алгоритму подаєтьсяпослідовністьвідліківділянки сигналу, щодосліджується на данійітерації. На цюпослідовністьнакладаєтьсяваговафункція і післяцьоговиконуєтьсядискретнеперетвореняФур'є.

Ваговафункціявикористовується для зменшенняспотворень в аналізіФур'є, яківикликаніскінченністюбібірки. На практиці в якостіваговоїфункції часто використовуютьсявікноХеммінга (1) та вікно Ханна (2).

—

де N — довжинавікнавиражена у відліках. ТодідискретнеперетворенняФур’єзваженого сигналу можназаписати у наступномувигляді:

—

Значенняіндексів k відповідаєнаступним частотам:

—

де F s — частота дискретизації сигналу.

Перехід в мелчастотну [3] область здійснюють за наступною формулою:

—

Нехай N_{FB} — кількістьфільтрів, (f_H, f_B) — досліджуванийдіапазончастот. Тодіцейдіапазонпереводятьвшкалумел, розбиваютьна N_{FB}рівномірнорозподіленихдіапазонитарозраховуютьвідповіднімежівобластіліній нихчастот. Позначимочерез$H_{m,k}$— ваговікоефіцієнтиотриманихфільтрів. ФільтризастосовуютьсядоквадратівмодулівкоефіцієнтівперетворенняФур’є. Отриманізначеннялогарифмуються:

Заключниметапом в розрахунку MFCC коефіцієнтів є дискретнекосинуснеперетворення:

Напрактицікількістькоефіцієнтів NMFCC дорівнює 12 (окрімпершого), оскількивонимістять 95% корисноїінформаціїпрозвуковийсигнал. [12]

3.3. Алгоритм роботи Adam

На Адама можна розглядати як поєднання RMSprop та стохастичного градієнтного спуску з імпульсом. Він використовує квадратичні градієнти для масштабування швидкості навчання, як RMSprop, і використовує імпульс, використовуючи ковзну середню градієнта замість самого градієнта, як SGD з імпульсом.

Адам - це адаптивний метод швидкості навчання, що означає, що він обчислює індивідуальні показники навчання для різних параметрів. Його назва походить від оцінки адаптивного моменту , і причиною, яку це називають, є те, що Адам використовує оцінки першого та другого моментів градієнта, щоб адаптувати швидкість навчання для кожної ваги нейронної мережі. N-й момент випадкової величини визначається як очікуване значення цієї змінної в степені n. Більш формально:

Де ,m-момент, X- випадкова змінна

Для оцінки моментів Адам використовує експоненціальноковзаючі середні, обчислені на основі градієнта, оціненого для поточної міні-партії:

Де m і v - ковзнісередні, g - градієнтпоточноїміні-партії, а бета - новівведені гіперпараметри алгоритму. Вони мають справді хороші значення за

замовчуванням 0,9 та 0,999 відповідно. Вектори ковзних середніх ініціалізуються нулями на першій ітерації.

Щоб побачити, як ці значення співвідносяться з моментом, визначеним як у першому рівнянні, розглянемо очікувані значення ковзних середніх. Оскільки m і v є оцінкамипершоготадругогомоментів, необхідна такавластивість:

Очікуванізначенняоцінювачівповиннідорівнюватипараметру, якийминамагаємосьоцінити, оскільки в нашомувипадкупараметромтакож є очікуванезначення. Якби ці властивості виконувались, це означало б, що ми маємо неупереджені оцінки . Тепер побачимо, що це не справедливо для наших ковзних середніх. Оскільки ми ініціалізуємо середні значення нулями, оцінювачі зміщуються до нуля. Доведемо, що для m (доведення для v буде аналогічним). Щоб довести, що нам потрібно сформулювати для m найперший градієнт. Спробуєморозгорнутипарузначень m, щобпобачити, якийшаблонмибудемовикористовувати:

Як бачите, чим далі ми розширюємо значення m, тим менше перших значень градієнтів сприяють загальному значенню, оскільки вони множаться на дедалі менші бета-версії. Захопившицейшаблон, миможемопереписатиформулудлянашогоковзногосереднього:

Тепердавайтеподивимосьнаочікуванезначення m, щобпобачити, яквоновідноситьсядосправжньогопершогомоменту, щобмимогливиправитиневідповідністьдвох:

У першому рядку використовуєтьсянова формула для ковзного середнього для розширення m. Далі наближаємо g [i] до g [t]. Тепер можена взяти його із суми, оскільки це зараз не залежить від i. Оскільки відбувається наближення, у формулі з’являється помилка C. В останньому рядку просто використовуємо формулу для суми кінцевого геометричного ряду. З цього рівняння слід зазначити дві речі.

Існує упереджений оцінювач. Це стосується не лише Адама, те саме стосується і алгоритмів, що використовують ковзні середні (SGD з імпульсом, RMSprop тощо).

Це не матиме великого ефекту, якщо це не початок тренування, оскільки значення бета до рівня t швидко йде до нуля.

Тепер потрібно виправити оцінювач, щоб очікуване значення було таким, яке необхідне. Цейкрокзазвичайназиваютькорекцією упередженості. Остаточніформулидляоцінювачабудутьтакими:

Єдине, щозалишилосязробити, - цескористатисяцимиковзнимисереднімидляіндивідуальногомасштабуваннярівнянавчаннядлякожногопараметра. Церобиться в Адамідужепросто, дляоновленнявагизробитинаступне:

Де w - ваги моделі, eta (схожа на букву n) - це розмір кроку (він може залежати від ітерації).

Висновки до розділу

В даному розділі був показаний основний алгоритм роботі розпізнавача команд управління. Основною частиною є попереднє навчання нейронної мережі, за допомогою розкладу вхідного набору звукових сигналів на окремі характеристики.

Дана модель дозволє після навчання отримати модель, готову до розпізнавання команд, що буде конвертована у окрему бібліотеку, що може використовуватися й у інших пристроях для розпізнавання команд з використанням плат Arduino.

РОЗДІЛ 4. ОПИС РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Написання програми

Процес розробки програми розділений на 2 етапи:

Перший – написання програми на мові Python, що дозволить отримати навчену модель, яка згодом буде конвертована у бібліотеку для середи розробки керуючої програми платою пристрою.

Для управління візком були виділені наступні 6 команд, для кожної з яких був створений клас, та окремий клас «шум»: вперед, назад, вліво, вправо, швидше, стоп.

Другий крок – написання керуючої команди, що буде працювати у інвалідній колясці та буде розпізнавати команди управління. Під час роботи керуюча команда буде записувати звук на вбудований мікофон, проводити класифікацію отриманого запису за допомогою раніше отриманої бібліотеки з навчаною нейронною мережею. Загальна схема роботи керуючої програми зображена на рис. 4.1.

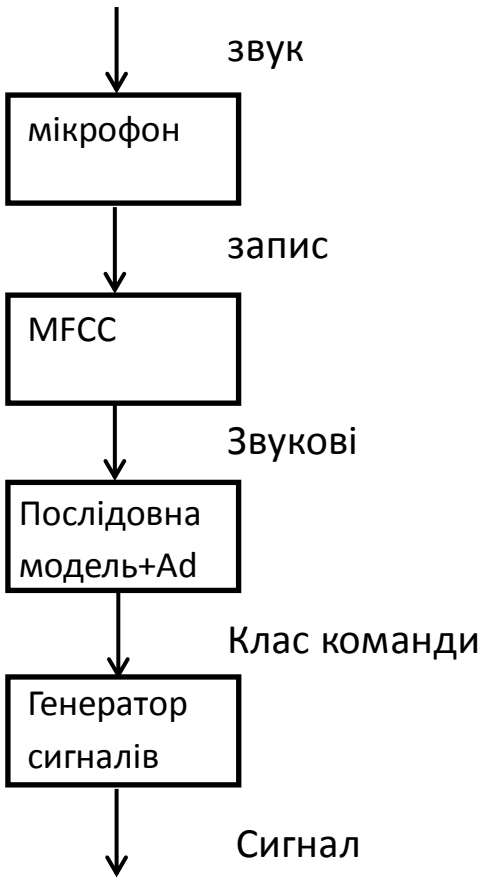

Рис. 4.1. загальна схема роботи керуючої програми

Блоки MFCC мають наступні характеристики:

- Довжина запису – 2 секунди
- Довжина вікна – 1200 мс
- Крок – 50 мс.

4.2. Результати тестування

В результаті тестування система показала свою працездатність. Були протестовані на розпізнавання усі 6 команд управління.

Під час тестування були випадки, коли програма неправильно розпізнавала команду, або не могла розбізнати команду. Загальний відсоток тоності розпізнавання, отриманий при тестуванні системи: 98%

Також були отимані й інші характеристики навчаноїнеронної мережі. Ці характеристики показані у таблиці 4.1.

Табл4.1. Результати експеременті

Кількість елементів у класі	Точність, %	Втрати	Час взаємодії, мс	Максимальна задіяна оперативна пам’ять, kB	Використання постійної пам’яті, kB
100	78.1	0.52	70	16	35.4

Як бачимо з таблиці, отимані результати характеризують систему к працездатну та єфективну, отримані дані дозволяють говорити про те, що можлива подальший розвиток системи.

Для більш детального опису отриманих характеристик були також протестовані ще 3 бібліотеки з іншими вхідними параметрами при навчанні нейронної мережі. Були також протестовані системі з іншою кількістю елементів у кожному класі, а також різною кількістю класів для повної картини впливу кількості класів на характеристики системи.

Табл 4.2. Розширені результати експерементів

Кількість класів	Кількість елементів у класі	Точність, %	Втрати	Час взаємодії, мс	Максимальна задіяна оперативна пам’ять, kB	Використання постійної пам’яті, kB
7	100	78.1	0.52	70	16	35.4
7	15	80,5	0.42	67	18,3	38,7
3	100	81.1	0.45	66	12	29.9
3	15	92.3	0.23	55	15.6	35.5

Висновки до розділу

Таким чином, розроблене рішення дозволяє розпізнавати дві голосові команди, використовуючи заздалегідь навчену мережу, що в подальшому використовується на платі з обмеженими ресурсами. Основною перевагою є використання власної вибірки, що дозволяє налаштувати систему під користувача. Додатковою перевагою є те, що система використовує невелику але сучасну плату, що здатна легко взаємодіяти з іншими платами того ж сімейства, що забезпечує універсальність та масштабованість. Система на практиці показала свою працездатність згідно з статистичними характеристики. В майбутньому система може бути покращена за рахунок збільшення навчальної вибірки, що збільшить точність розпізнавання, а також за рахунок збільшення словника, що позитивним чином вплине на практичність системи.

РОЗДІЛ 5. СТАРТАП

5.1. Опис ідеї проекту

При описі ідеї проєкту були отримані напрямки застосування та вигоди для користувача, згідно із поставленою задачею були отримані наступні ідеї:

Табл.5. 1. Опис ідеї стартап-проекту

Зміст ідеї	Напрямки застосування	Вигоди для користувача
	1. Використання для інвалдного візку	Користувачу не потрібно використовувати руки для керування візком

Серед потенційних техніко-економічних переваг можна вказати:

- використання сучасних апаратних засобів,
- використання української мови,
- повна автономність виробу,
- використання індивідуальної вибірки для навчання системи

Згідно зотриманими параметрами побудована таблиця характеристик проекту

Табл.5. 2. Визначення сильних, слабких та нейтральних характеристик

№ п/п	*Техніко-економічні характеристики ідеї*	*(потенційні) товари/концепціїконкурентів*			*W (слабка сторо*	*N (нейтральна сторона)*	*S (сильна сторо*
		Мій проект	VoiceRecognitionModule	BTVOICECONTROL			

					на)		*на)*
1	Сучасні апаратні засоби	X			S		
2	Українська мова	X			S		
3	автономність	X	X		N		
4	Індивідуальна вибірка	X		X	N		

5.2. Технологічний аудит ідеї проекту

Згідно з описанимивищеідеми проекту були описані та зібрані у таблицю технології реалізації

Табл.5.3. Технологічна здійсненність ідеї проекту

№ п/п	*Ідея проекту*	*Технологіїїреалізації*	*Наявністьтехнологій*	*Доступністьтехнологій*
1	1	Використання електричного інвалідного візку з заміною блоку управління на плату Arduino	Дана технологія наявна	Не доступна
2	1	Використання звичайного інвалідного візку з подальшою його електрифікацією	Дана технологія повинна бути розроблена	Не доступна
Обрана технологія реалізації ідеї проекту: 1				

Згідно з таблицею можна зробити висновок що оптимальною технологіє буде модифікація електричного візка.

5.3. **Аналіз ринкових можливостей запуску стартап-проекту**

Враховуючи, що даний проект має дуже вузьку спеціалізацію і конкретну групу людей, для яких він буде використовуватись, можна передбачити низьку конкуренцію, але натомість і низький попит

5.3.1. Аналіз попиту: наявність попиту, обсяг, динаміка розвитку ринку

Задля наглядніщогоаналіщу, побудуємо таблицю попередньої характеристики потенційного ринку стартап-проекту

Табл.5.4. Попередня характеристика потенційного ринку стартап-проекту

№ п/п	*Показники стану ринку (найменування)*	*Характеристика*
1	Кількістьголовнихгравців, од	1
2	Загальнийобсяг продаж, грн/ум.од	140000 грн.
3	Динаміка ринку (якіснаоцінка)	Стагнує
4	Наявністьобмежень для входу (вказати характер обмежень)	Немає обмежень
5	Специфічнівимоги до стандартизації та сертифікації	Сертификат медичного обладнання
6	Середня норма рентабельності в галузі (або по ринку), %	20%

Згідно з таблицею можемо зробити висновок, що ринок не є привабливим для входження.

5.3.2. Визначення потенційних груп клієнтів

Існує дві групи клієнтів, що можуть бути потенційними елієнтами стартап-проектую

Табл.5.5. Характеристика потенційних клієнтів стартап-проекту

№ п/п	*Потреба, щоформуєринок*	*Цільова аудиторія*	*Відмінності у поведінці різних*	*Вимогиспоживачівдо товару*

		(цільові сегменти ринку)	*потенційних цільових груп клієнтів*	
1	Підвищення мобільності для людей з обмеженими можливостми	Люди з обмеженими можливостями	Головний фактор – зручність длявикористання	Надійність як товару так і постачальника
		Клініки	Купуються для підвищення іміджу закладу	Надійність як товару так і постачальника

5.3.3. Аналізринковогосередовища

Наведемо основні фактори загроз та можливостей

Табл.5.6. Фактори загроз

№ п/п	*Фактор*	*Змістзагрози*	*Можливареакціякомпанії*
1	Зменшення покупної спроможності	Зменшення попиту	Оптимізація виробництва та зменшення вартості
2	Вихід на ринок кращого продавця	Зменшення попиту	Владування коштів у рекламу компанії

Табл.5.7. Фактори можливостей

№ п/п	*Фактор*	*Змістможливості*	*Можливареакціякомпанії*
1	Збільшення покупної спроможності	Збільшення попиту	Збільшення обсягів виробництва
2	Поліпшення настроїв щодо роботизованих систем	Розширення групи клієнтів	Більш активне просування продукції

5.3.4. Аналіз пропозиції

Враховуючи, що важливою характеристикою є те, що даний проект надає можливість використовувати українську мову, можна встановити наступні особливості конкурентного середовища:

Табл.5.8. Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	*В чому проявляється дана характеристика*	*Вплив на діяльність підприємства (можливі дії компанії, щоб бути конкурентоспроможною)*
1. Вказати тип конкуренції - чиста	Ринок не має великих фінансових обертів	Покращення репутації
2. За рівнем конкурентної боротьби - національний	Основою є використанн української мови	Просування ідеї у маркетингу
3. За галузевою ознакою - внутрішньогалузева	Можлва конкуренція лише з товарами того ж виду	Покращення кості
4. Конкуренція за видами товарів: - товарно-видова	Можлва конкуренція лише з товарами того ж виду	Зниження ціни
5. За характером конкурентних переваг - нецінова	Товар є специфічним, тому більшу значимість має функціональність	Збільшення ціни
6. За інтенсивністю - не марочна	Товар має у собі нові технології	Ровинення технологій

5.3.5. Більш детальний аналіз умов конкуренції в галузі

Згідно з попереднім пунктом проведемо більш детальний аналіз.

Табл.5.9. Аналіз конкуренції в галузі за М. Портером

Складові аналізу	*Прямі конкуренти в галузі*	*Потенційні конкуренти*	*Постачальники*	*Клієнти*	*Товари-замінники*

	Коляска МІТ	*Здобуття інформованості населенно щодо продукції*	*Необхідні потачальники електричних візків та плат*	*Покупці з високою покупною спроможністю*	*Більша функціональність*
Висновки:	Невисока інтенсивність	Є можливість входу при відсутності потенційних конкурентів	Постачальник зможе диктувати ціни на ринок за рахунок монополії на необхідні плати	Клієнти не диктують умови роботи	Незначні обмеження

Сильною стороною проекту є його унікальність: використанн голосових команд у купі з використанням української мови.

5.3.6. Обґрунтування переліку факторів конкурентоспроможності

Згідно з вказаними вище характеристиками виділяється незначна кількість факторів конкурентоспроможності, але вони мають великий вплив на фактор вцілому.

Табл.5.10. Обґрунтування факторів конкурентоспроможності

п/п	*Фактор конкурентоспроможності*	*Обґрунтування (наведення чинників, що роблять фактор для порівняння конкурентних проектів значущим)*
	Використання укріїнської мови	Не існує інвалідних візків з голосовим управлінням на українській мові
	Індивідуальне налаштування	Індивідуальний підхід є значущим для характеристики точності розпізнавання

5.3.7. Аналізсильних та слабкихсторін проекту

Табл.5.11. Порівняльний аналіз сильних та слабких сторін «назва проекту»

п/п	*Фактор конкурентоспроможності*	*Бали 1-20*	*Рейтинг товарів-конкурентів у порівнянні з … (назва підприємства)*						
			3	*2*	*1*		*1*	*2*	*3*
	Використання укріїнської мови	20							

	Індивідуальне налаштування	15							

5.3.8. SWOT-аналіз

Підсумуємо вище сказане, та побудуємо матрицю SWOT-аналізу

Табл.5.12. SWOT- аналіз стартап-проекту

Сильні сторони: Використанняукріїнськоїмови; Індивідуальненалаштування	Слабкі сторони: Залежність від постачальника плат
Можливості: Збільшення покупної спроможності; Полібшення настроїв щодо роботизованих систем	Загрози: Зменшення покупної спроможності; Вихід на ринок кращого продавця

5.3.9. Альтернативи ринкової поведінки

Визначемо альтернативи ринкової поведінки

Табл.5.13. Альтернативи ринкового впровадження стартап-проекту

п/п	*Альтернатива (орієнтовний комплекс заходів) ринкової поведінки*	*Ймовірність отримання ресурсів*	*Строки реалізації*
	Зміна виробника плат	Висока	6 місяців
	Зміна групи покупців	Висока	2 роки

Отже альтернативою ринкової поведінки є зміна постачальника плат для проекту.

5.4. Розроблення ринкової стратегії проекту

У наступних підпунктах буде розгянута розробка ринкової стратегії проекту

5.4.1. Опис цільових груп потенційних споживачів

Табл.5.14. Вибір цільових груп потенційних споживачів

п/п	*Опис профілю цільової групи потенційних клієнтів*	*Готовність споживачів сприйняти продукт*	*Орієнтовний попит в межах цільової групи (сегменту)*	*Інтенсивність конкуренції в сегменті*	*Простота входу у сегмент*
	Інивідуальні покупці	Середня	Низький	Низька	Висока
	Підприємці	Висока	Низький	Нищька	Низька
Які цільові групи обрано: Індивідуальні покупці (люди з обмеженими можливостями)					

Отже вибором є стратегія концентрованого маркетингу

5.4.2. Базова стратегія розвитку

Визначемо базову стратегію розвитку

Табл.5.15. Визначення базової стратегії розвитку

п/п	*Обрана альтернатива розвитку проекту*	*Стратегія охоплення ринку*	*Ключові конкурентоспроможні позиції відповідно до обраної альтернативи*	*Базова стратегія розвитку**
	Зміна виробника плат	стратегія концентрованого маркетингу	Використання української мови	Стратегія спеціалізації

5.4.3. Вибірстратегіїконкурентноїповедінки

Табл.5.15. Визначення базової стратегії розвитку

п/п	*Чи є проект «першопрохідцем» на ринку?*	*Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?*	*Чи буде компанія копіювати основні характеристики товару конкурента, і які?*	*Стратегія конкурентної поведінки**
	так	Пошук нових	Характеристи товарів конкурента не копіюються	Стратегія лідера

5.4.4. Стратегія позиціонування

Визначемо стратегії позиціонування

Табл.5.17. Визначення стратегії позиціонування

п/п	*Вимоги до товару цільової аудиторії*	*Базова стратегія розвитку*	*Ключові конкурентоспроможні позиції власного стартап-проекту*	*Вибір асоціацій, які мають сформувати комплексну позицію власного проекту (три ключових)*
	Висока покупна спроможність	Змен шення цін	Використання не дорогих систем	Вигода бренду
	Інтерес у продукті	Рекла мна кампанія	Використання сучасних технологій	Відміни від конкурентів

5.5. Розроблення маркетингової програми стартап-проекту

У наступних підпунктах будуть розгянуті складові маркетингової програми стартап проекту.

5.5.1. Маркетингова концепція товару

Табл.5.18. Визначення ключових переваг концепції потенційного товару

п/п	*Потреба*	*Вигода, яку пропонує товар*	*Ключові переваги перед конкурентами (існуючі або такі, що потрібно створити*
	Надійніст ь	Індивідуаль не налаштування	Включення фактору нечіткої вимови при експлуатації товару
	Легкість у	Використанн	Використання зручної для покупця мови

	використанні	я української мови	

5.5.2. Маркетингова модель товару.

Табл.5.19. Опис трьох рівнів моделі товару

Рівні товару	*Сутність та складові*		
I. Товар за задумом	Основна потреба – підвищення мобільності людей з обмеженими можливостями, основні функціональні вигоди: індивідульний підхід, українська мова		
II. Товар у реальному виконанні	Властивості/характеристики	М/Нм	Вр/Тх /Тл/Е/Ор
	1. Надійність	М	Тх
	2. Зручніть	М	Тх
	Якість: перевірка точності розпізнавання		
	Пакування при необхідності для покупця		
	Марка: Ізіго + АКДІ1.1		
III. Товар із підкріпленням	До продажу		
	Після продажу		
За рахунок чого потенційний товар буде захищено від копіювання: використання закритих технологій			

5.5.3. Визначення цінових меж встановлення ціни

Табл.5.20. Визначення меж встановлення ціни

п/п	*Рівень цін на товари-замінники*	*Рівень цін на товари-аналоги*	*Рівень доходів цільової групи споживачів*	*Верхня та нижня межі встановлення ціни на товар/послугу*
	80000грн	30000грн	20-40 тис.	60000/100000

5.5.4. Оптимальна система збуту

Виведемо таблицею оптимальну систему збуту, в межах кого приймається рішення

Табл.5.21. Формування системи збуту

п/п	*Специфіка закупівельної поведінки цільових клієнтів*	*Функції збуту, які має виконувати постачальник товару*	*Глибина каналу збуту*	*Оптимальна система збуту*
	Низький попит	Швидкість та акуратність	Невелика	Від виробника до клієнта

5.5.5. Розроблення стратегії маркетингових комунікацій

Табл.5.22. Концепція маркетингових комунікацій

п/п	*Специфіка поведінки цільових клієнтів*	*Канали комунікацій, якими користуються цільові клієнти*	*Ключові позиції, обрані для позиціонування*	*Завдання рекламного повідомлення*	*Концепція рекламного звернення*
	Збільшення інтересу у роботизованих системах	Інтернет, телевізор	Мережа інтернет	Збільшення кількості покупців	Нові розробки
	Зібльшення покупної спроможності	Інтернет, Телевізор	Мережа інтернет	Збільшення кількості покупців	Нові розробки

Висновки до розділу

Згідно з зазначеними у розділі даними можна зробити висновок, що існує можливість инкової комерціалізації проекту, адже наявний попит, хоча і з низькою динамікою ринку та рентабельністю роботи на ньому.

Існує перспектива впровадження з огляду потенційної групи клієнтів. Бірьери для входження є низькими, а стан конкуенції є позитивним.

Доцільно альтернативою обрати для ринкової реалізації виріб з використанням інших плат для усунення ризику монополії на їх постачання

Подальша імплементація проекту є доцільною.

ВИСНОВКИ

Постійний розвиток інформаційних технологій призводить до збільшення кількості існуючих апаратних та програмних засобів для використання їх у побуті. Це надає можливість імплементації сучасних технологій у будь-які сфери соціуму. Це призводить до того, що навіть обмежені групи людей можуть покращити свій побут за рахунок використання сучасних технологій, а саме за допомогою новітніх систем та алгоритмів для розпізнавання мови з’являється можливість для створення систем розпізнавання команд управління для інвалідних візків.

В результаті виконання магістерської дисертації:

- Були досліджені існуючі розпізнавачі команд на базі Arduino. Були переразовані їх основні характеристики та принципи роботі, особливості роботи, та області застосування.
- Була проведена порівняльна характеристика існуючих рішень, та була обгрунтована необхідність створення їх аналогу.
- Були виділені основні функції системи розпізнавання, серед яких е використання сучасних засобів, вікористання української мови для словника нейронної мережі, а також повна автоматичність для використання системи у інвалідних візках.

- Був описаний алгоритм, а такой математично обгрунтований підхід до розв’зання поставленої задачі.
- Описана розробка та приведені результати тестування з наведенням вихідних характеристик системи. Також були наведені порівнльні характеристики данної системи з декількома варіантами вхідних данних при навчанні мережі.

СПИСОК ЛІТЕРАТУРИ

1. IntroductiontoVoiceRecognitionWithElechouse V3 andArduino [електронний ресурс] – Режим доступу: https://www.instructables.com/Introduction-to-Voice-Recognition-With-Elechouse-V/

2. ГолосовоеуправлениепосредствомArduino [електронний ресурс] – Режим доступу: https://future2day.ru/golosovoe-upravlenie-posredstvom-arduino/

3. MIT IntelligentWheelchair Project: A Voice-CommandableRoboticWheelchair [електронний ресурс] – Режим доступу: https://techtv.mit.edu/videos/6465-mit-intelligent-wheelchair-project-a-voice-commandable-robotic-wheelchair

4. Metz, Cade (November 9, 2015). "GoogleJustOpenSourcedTensorFlow, ItsArtificialIntelligenceEngine". Wired. RetrievedNovember 10, 2015.

5. Piatetsky, Gregory. "Pythoneatsawayat R: TopSoftwareforAnalytics, DataScience, MachineLearningin 2018: TrendsandAnalysis". KDnuggets. KDnuggets. Retrieved 30 May 2018.

6. A. Y. Ngand M. I. Jordan. OnDiscriminativevs. GenerativeClassifiers: A comparisonoflogisticregressionandNaiveBayes. in NIPS 14, 2002.

7. Ben-Hur, Asa; Horn, David; Siegelmann, Hava; Vapnik, Vladimir N. ""Supportvectorclustering" (2001);". JournalofMachineLearningResearch. 2: 125–137.

8. R. Quinlan, "Learningefficientclassificationprocedures", MachineLearning: anartificialintelligenceapproach, Michalski, Carbonell&Mitchell (eds.), MorganKaufmann, 1983, p. 463–482.

9. Altman, Naomi S. (1992). "Anintroductiontokernelandnearest-neighbornonparametricregression" (PDF). TheAmericanStatistician. 46 (3): 175–185.

10. N. Qian, “Onthemomentumtermingradientdescentlearningalgorithms.” NeuralNetworks : TheOfficialJournaloftheInternationalNeuralNetworkSociety, 1999, 12(1), 145–151.

11. Дідерік П. Кінгма та Джиммі Лей Ба. Адам: Метод стохастичної оптимізації . 2014

12. АНАЛІЗ ВПЛИВУ ПАРАМЕТРІВ ОБРОБКИ ЗВУКОВОГО СИГНАЛУ НА ЯКІСТЬ РОЗПІЗНАВАННЯ ГОЛОСОВИХ КОМАНД. // Вісник Національного технічного університету України «КПІ». – 2014. – С. 34–41.